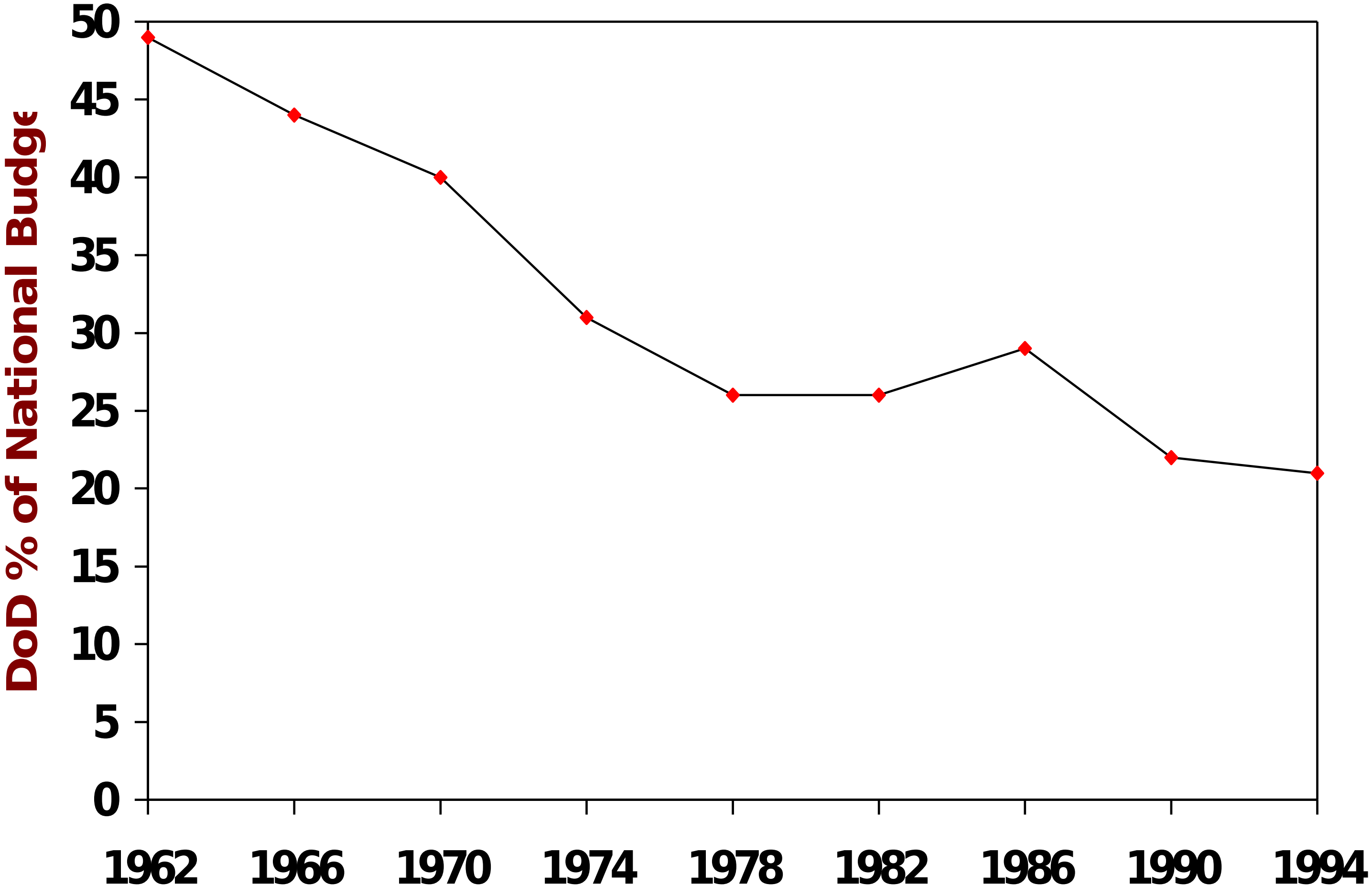
Software Estimating
Modules 9 & 10

"Let's Take it From the Top"

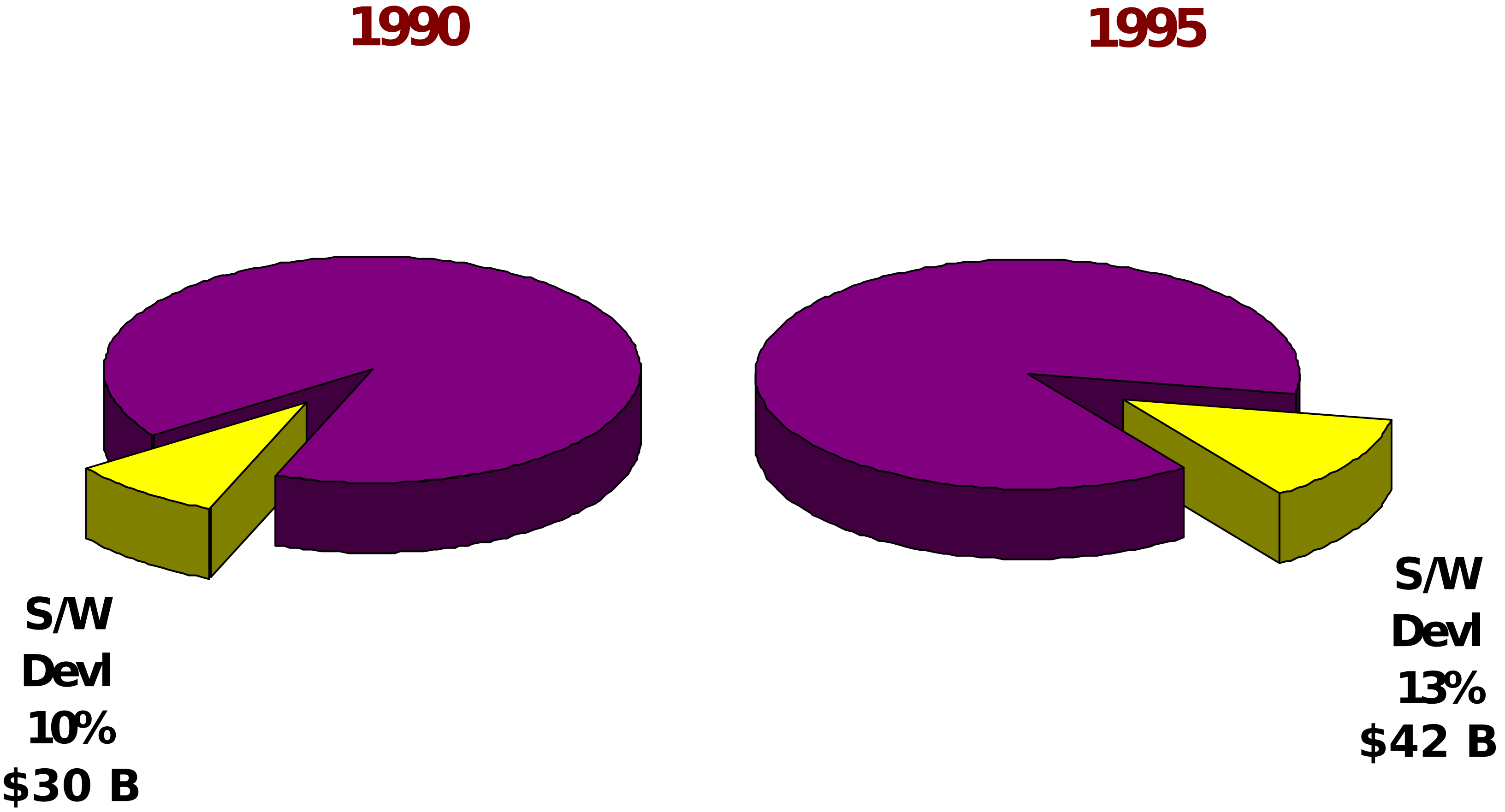
ESC Cost Core Training
Developed By

USAF ESC/FMC
Hanscom AFB, MA
Apr 91
by ESC/FMC
Peg Wells

DoD's Percent of the National Budget



Software Development in the DoD Budget



Software Cost Estimating Contents

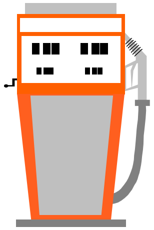
- ✓ ***Overview of Hardware & Software***
- Steps of a Software Cost Estimate
- SEER SEM
- Specific to ESC
- Common Mistakes
- Current Issues & Conclusions

Overview of H/W & S/W Definitions

Program: Sequence of instructions designed to perform a task

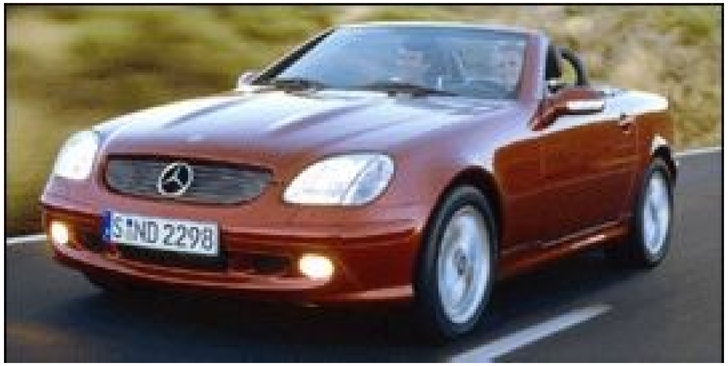
Software: A program or set of programs that control a computer system

Overview of H/W & S/W Which is More Important?



Software

Hardware



Both are equally important!

They work together to produce a thinking machine.

Overview of H/W & S/W Concepts (Cont.)

- Analog Computers
  - Continuous Electrical Signal
  - Slow and Outdated
- Digital Computers
  - Sequence of Electronic Signals
  - Faster & less prone to distortion
    - Binary Digits (BIT)
    - BYTE = 8 BITS
    - Word varies in size
      - How information is stored
      - Larger word size = Faster processing and access to a larger memory

Overview of H/W & S/W

ESC/FMC H/W Timeline Example

	1984	1988	1991	1993	1995	2000
	Z-100	Z-248	Unysis	Dell	IBM	PC/ Notebook
Processor	8085/8088	286	386	486	Pentium	Pentium II
Speed		8-10 MgHz	20-25 MgHz	33-50 MgHz	90 MgHz	450 MgHz
RAM	.192 Mgs	2-4 Mgs	4 Mgs	8-16 Mgs	16 Mgs	128 Mgs
Hard Drive	<10 MB	20 MB	100 MB	220 MB	850 MB	17 GB
Monitor	Mono	EGA	VGA	SVGA	SVGA	SVGA
	$2250	$1628	$1800	$3075	$3300	$3500

Including Software

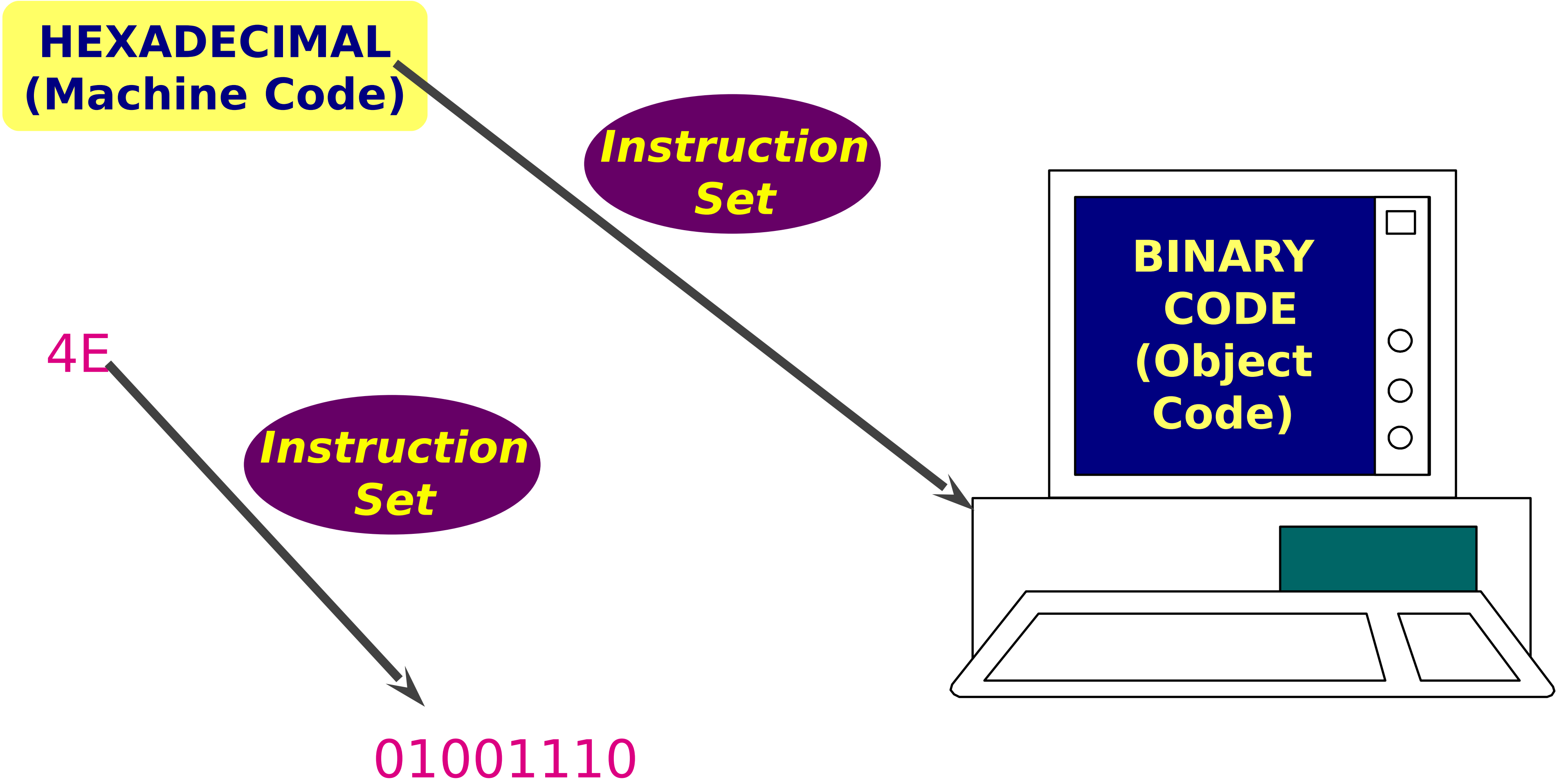
386 Upgrades
20 MgHz
100 MB
$1500

Computers are outdated approximately every 18 months.

Overview of H/W & S/W
ESC/FMC S/W Timeline Example

	1986	1988	1991	1993	1995	2000
Oper Sys	DOS 1.1	DOS 3.1	DOS/WIN	WIN 3.1	WIN 95	WIN 2000
Comm	VAX (DEC)	VAX (CALL)	VAX (ZSTEM)	BMAIL SERVER	BMAIL SERVER	MS Outlook NT/SERVER
Word Proc	CPT, Peach Text	Enable, Word Perf	Word Perfect	Word	Word	Word
Sprdsht	Lotus 1.0	Lotus, Enable	Lotus 2.1, Enable	Excel	Excel	Excel
Dsktop Pub	Graphics Lab	Chart, Harv Graphics	Harv Grph, Pixie	Power Point	Power Point	Power Point

Overview of Software Programming Languages



HEXADECIMAL (Machine Code)

Instruction Set

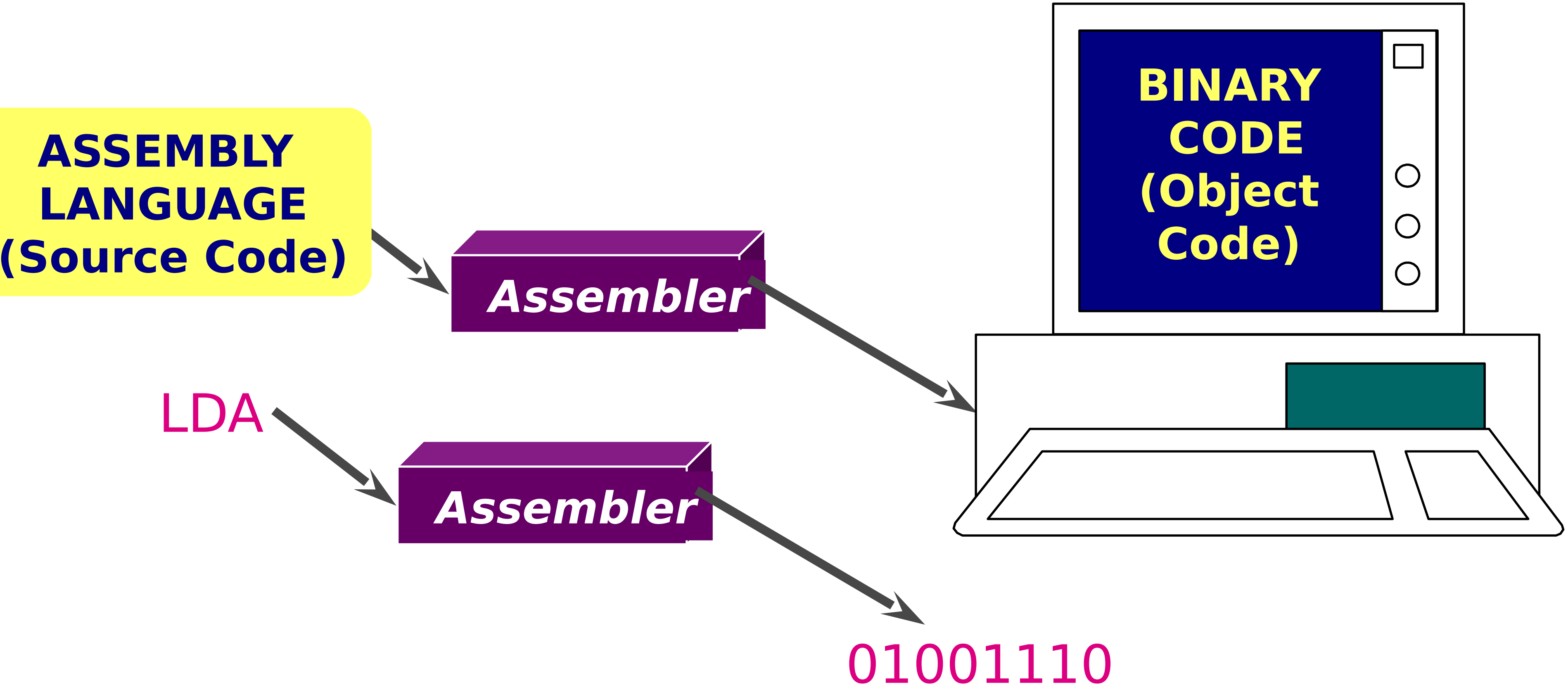
BINARY CODE (Object Code)

4E

Instruction Set

01001110

Overview of Software Programming Languages



ASSEMBLY LANGUAGE (Source Code)

Assembler

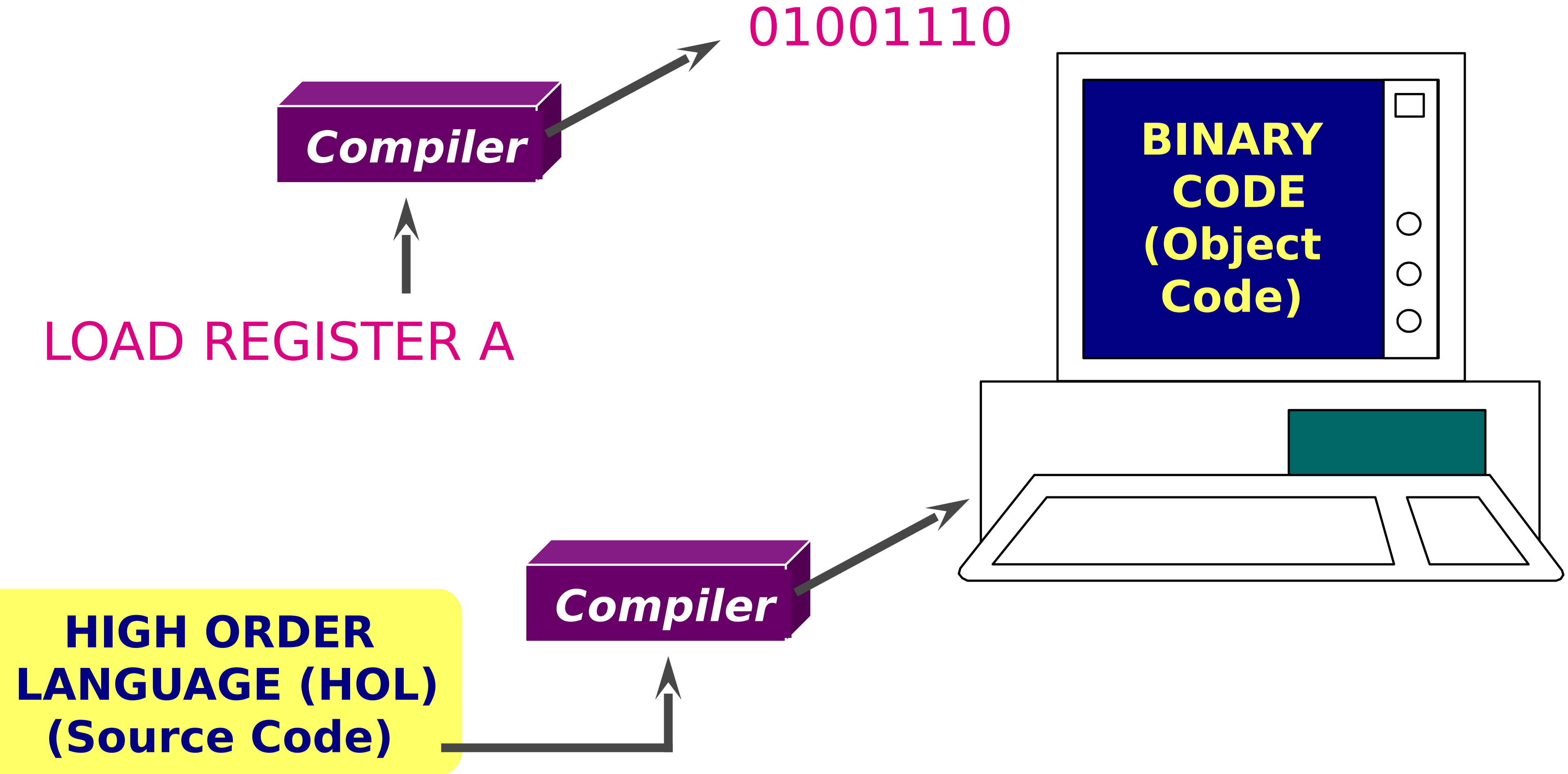
BINARY CODE (Object Code)

LDA

Assembler

01001110

Overview of Software Programming Languages



01001110

Compiler

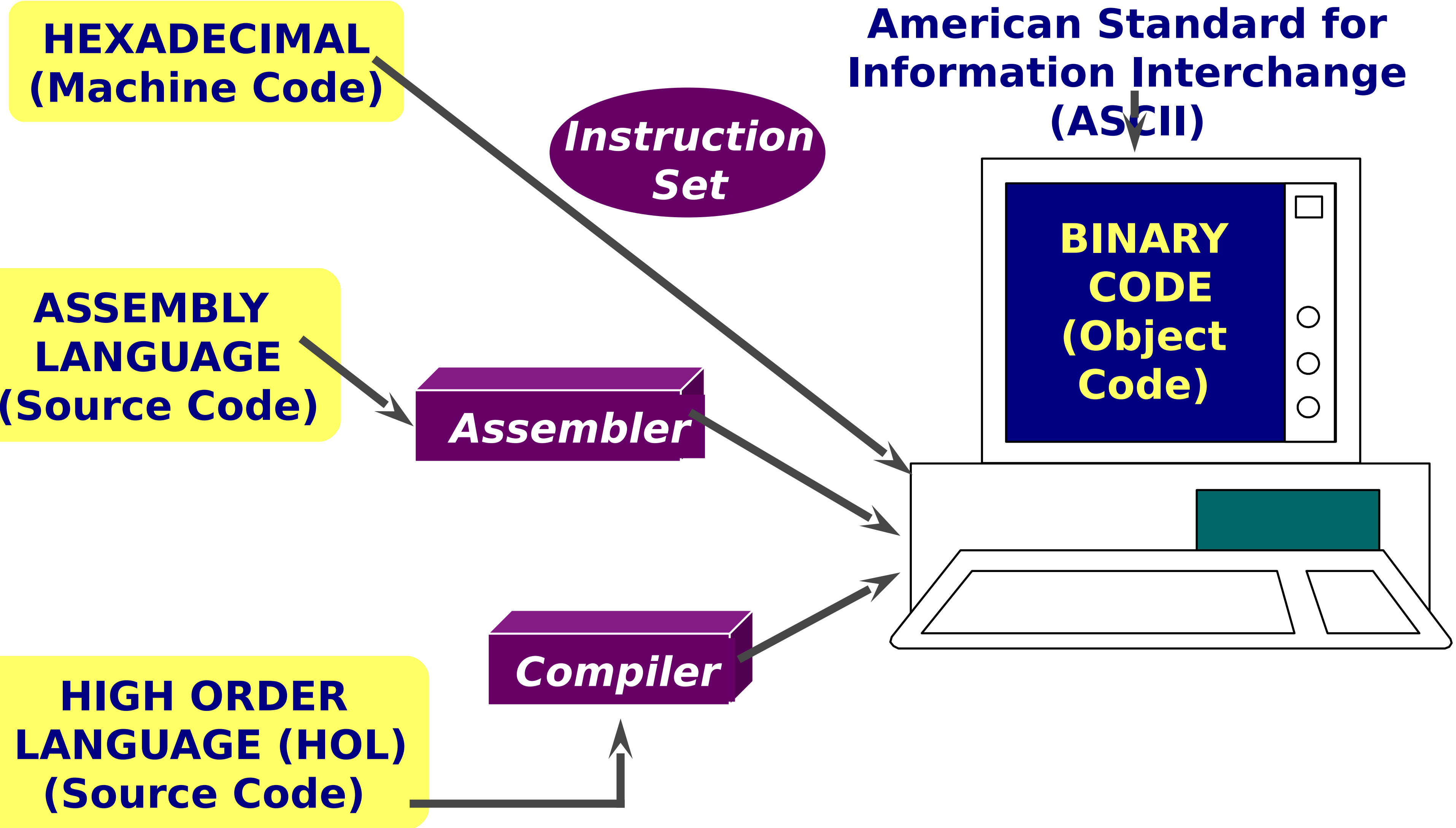
LOAD REGISTER A

BINARY CODE (Object Code)

Compiler

HIGH ORDER LANGUAGE (HOL) (Source Code)

Overview of Software Programming Languages



HEXADECIMAL (Machine Code)

Instruction Set

American Standard for Information Interchange (ASCII)

BINARY CODE (Object Code)

ASSEMBLY LANGUAGE (Source Code)

Assembler

Compiler

HIGH ORDER LANGUAGE (HOL) (Source Code)

Overview of Software Programming Languages

- FORTRAN **(Formulated Translation)**
  - 1950’a: Scientific Software
  - Outdated
- CoBOL **(Common Business Oriented Language)**
  - 1950’s Business Software
  - Somewhat outdated
- Basic **(Beginners’ All-Purpose Symbolic Instruction Code)**
  - 1960’s: Educational & Personal Computers
  - Easy to Learn, but slow and clumsy

Overview of Software Programming Languages

- Pascal (PL/1)
  - 1960’s: First Structured Programming Language
  - Better organized & easier to read
  - PL/1, Jovial
- Jovial
- C, C++
- Ada
  - 1983, updated in 1995
  - Mandated Language for DoD mission critical software from 1983-1994
  - Developed by DoD

Overview of Software
ogramming Language Generatio

- **First Generation (11100101)**
  - Machine Language – so, machine dependent programming
  - Hard-wired instructions
  - Numeric Instructions and addresses

- **Second Generation (IBM BAL, Assembly)**
  - Machine-dependent programming
  - Translation of program with an assembler
  - Symbolic instructions and addresses

- **Third Generation (COBOL, FORTRAN, Pascal, Ada, C, Basic, PL/I)**
  - Problem-oriented languages
  - Translation with compilers or interpreters
  - Structured programming, database management systems

- **Fourth Generation**
  - Non-procedural languages
  - Integrated data dictionaries
  - Dynamic relational databases

- **Fifth Generation (PROLOG)**
  - Artificial Intelligence, fuzzy logic and neural networks

Overview of Software Programming Languages

FOURTH GENERATION LANGUAGES (4GL)

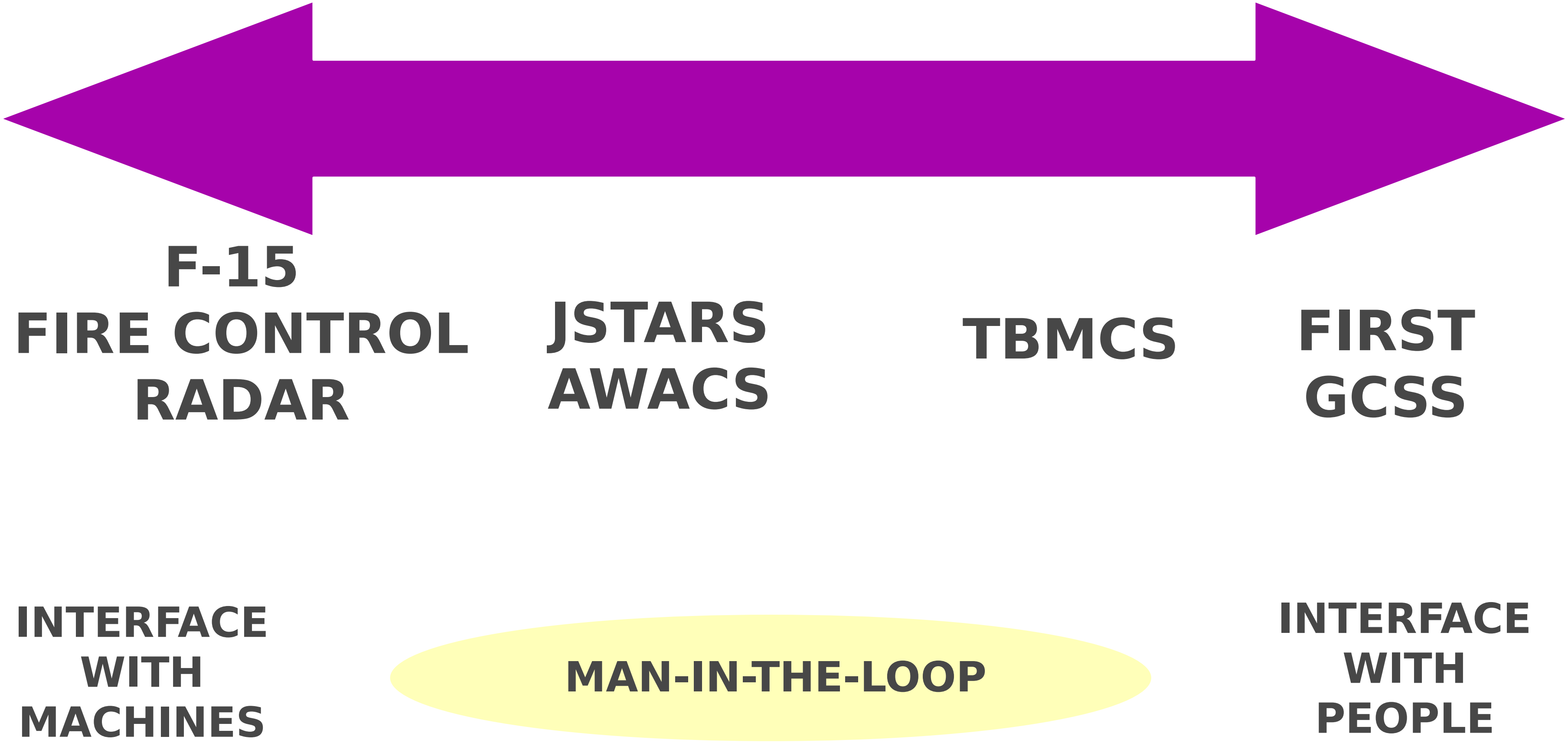
(Object Oriented Language)

Examples: Structured Query Language (SQL)
Lotus
Excel
Oracle

Drawbacks:

- Compilers are not efficient
  - Process very slow
  - Use large amounts of memory
- Standardization is not possible

Overview of Software
Software Spectrum



F-15
FIRE CONTROL
RADAR

JSTARS
AWACS

TBMCS

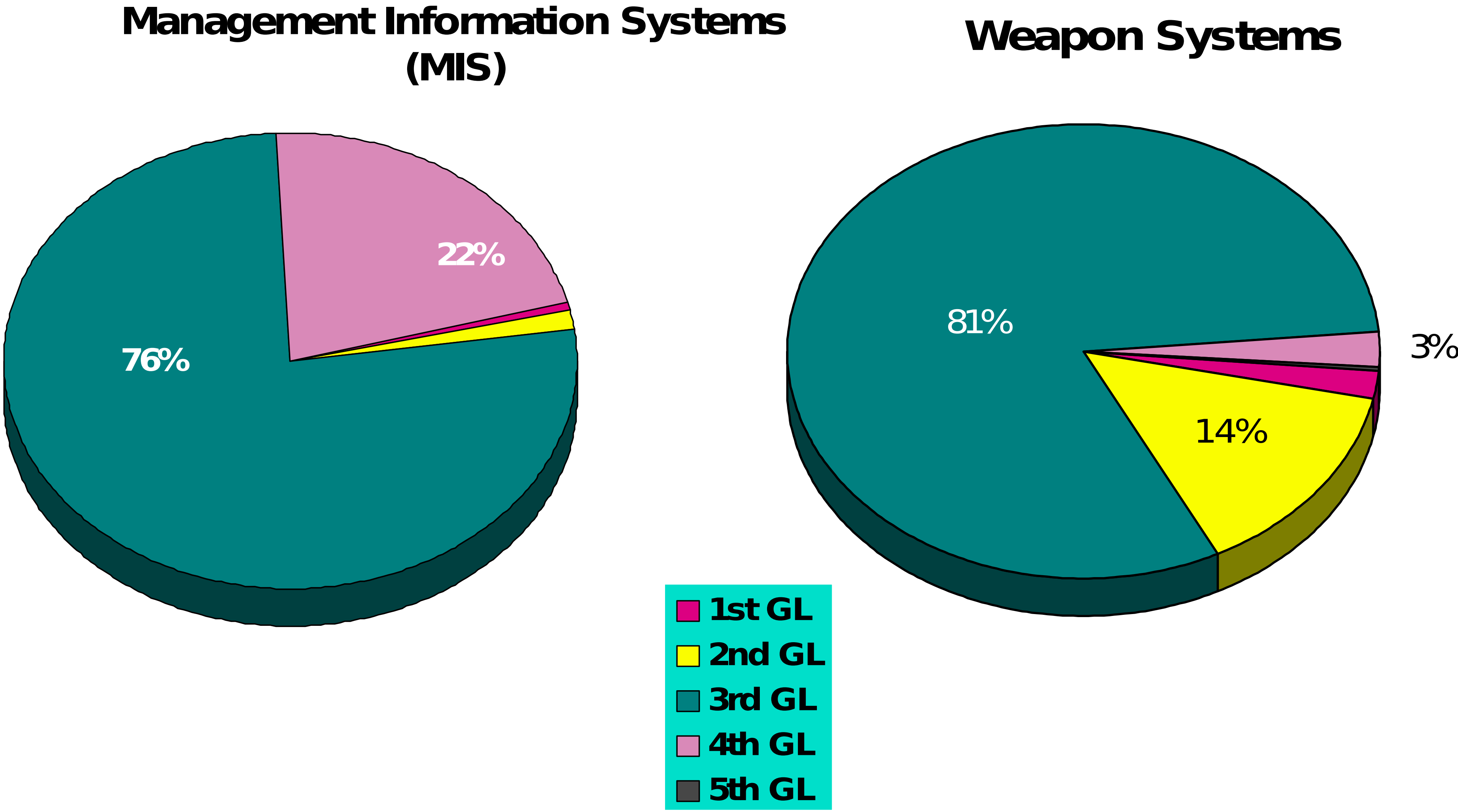
FIRST
GCSS

INTERFACE
WITH
MACHINES

MAN-IN-THE-LOOP

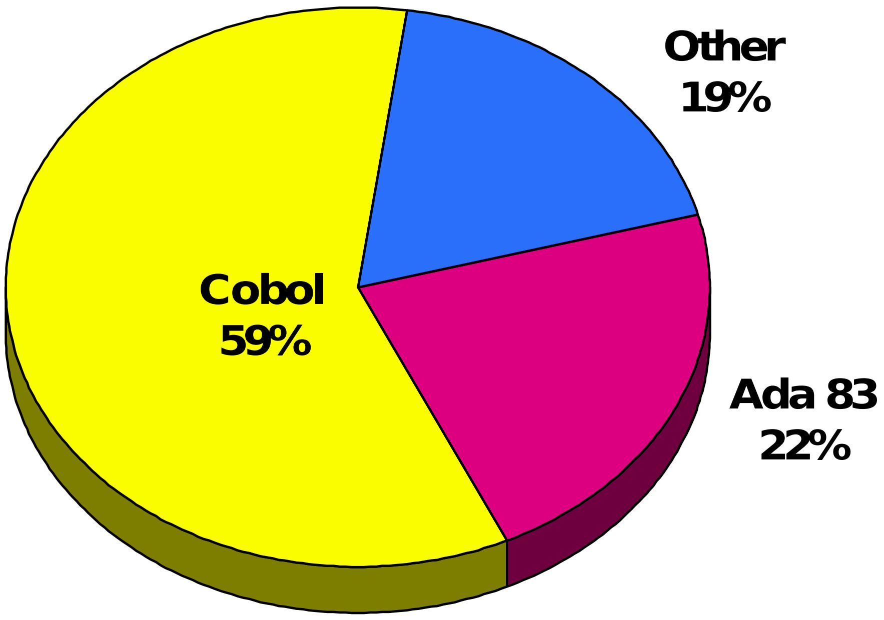
INTERFACE
WITH
PEOPLE

Overview of Software Language Generations

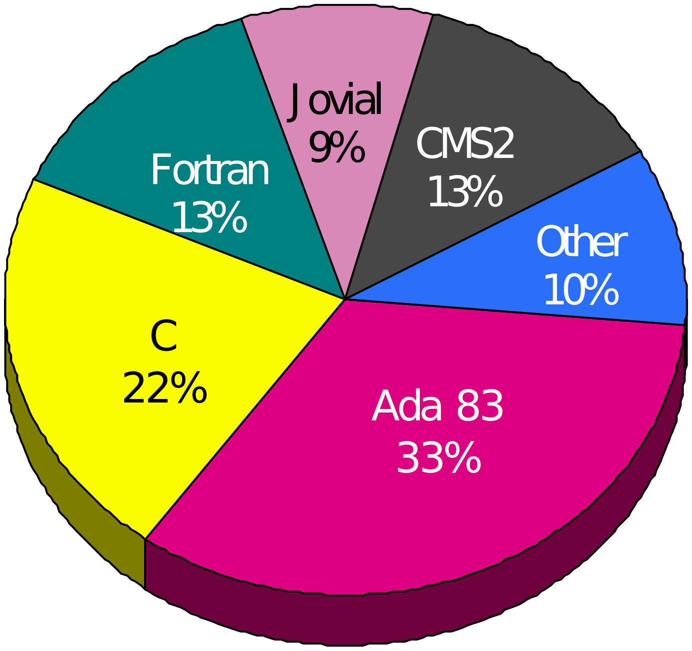


Overview of Software HOL (3GL) Types

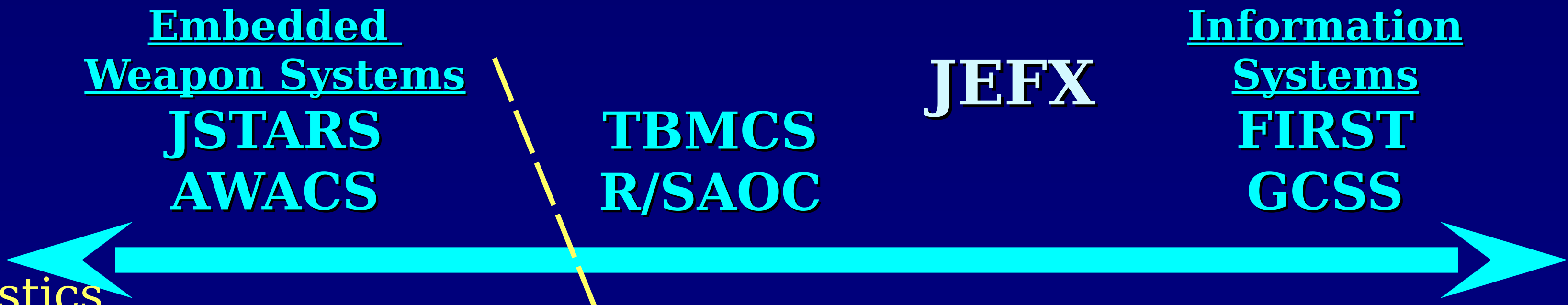
Management Information Systems (MIS)



Weapon Systems



Environment of Cost Estimating



Embedded Weapon Systems

JSTARS
AWACS

TBMCS
R/SAOC

JEFX

Information Systems

FIRST
GCSS

Characteristics

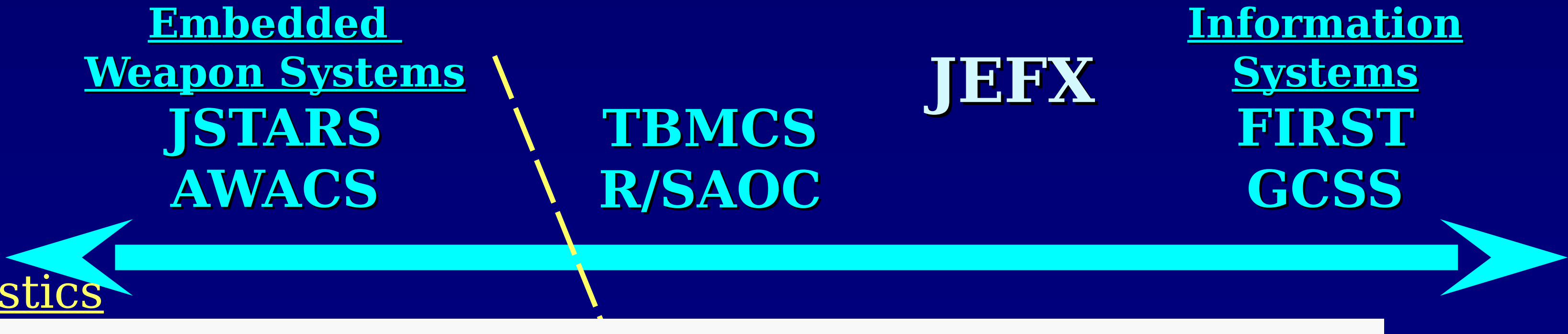
Technical	-Devl H/W -Devl S/W (3rd Generation Lang.)	-COTS H/W & S/W -Devl S/W (4th Generation Lang.)
Estimating		
Hardware	- Analogies, CERs	- Vendor Quotes
Software	- Lines of Code Models	- # of Forms, Screens, etc.
Integration	- Hardware & Software	- COTS S/W (Glue Code) - Rapid Integration

Cost Tools

Tools Available
Historical Data is Available
Need annual verification & validation
Requires maintenance & updating

Limited Tools Availabl
Need to develop more
i.e: IT S/W CERs

Environment of Cost Estimating



Characteristics

Tec

Es

F

S

I

Co

g.)

reens,

ue

Availabl

ion

p more

ERs

Goal: New tools developed - open environment

Moving from left to right on the previous chart

- Current available tools are not good fit

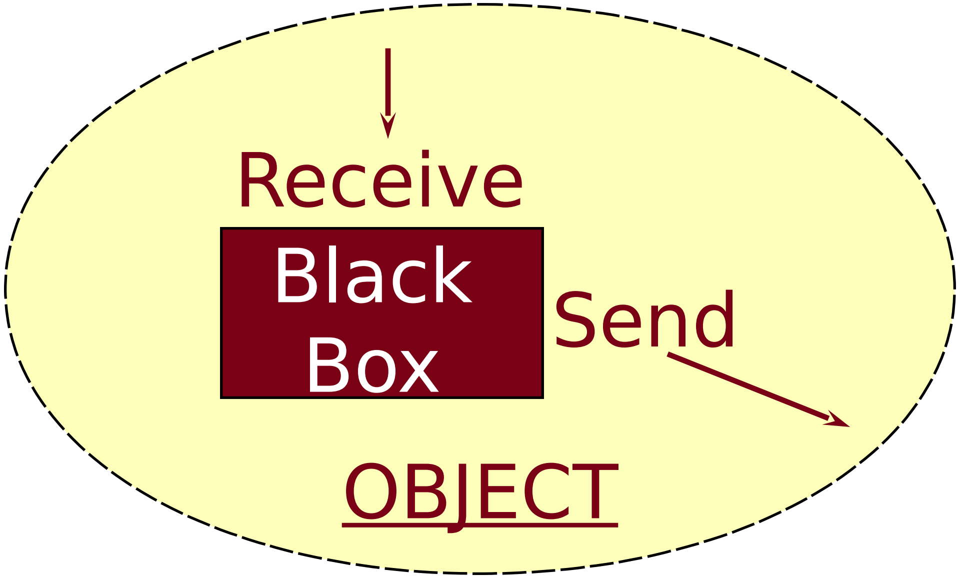
- Historical data from Embedded Weapon Systems is available

Estimating Characteristics:

- **S/W** -- Commercial Models are Lines of Code derived

-- AIS development (4th Generation Languages and Object Oriented Development) LOC not applicable

Overview of Software
Object Oriented



Examples:
- C++
- Smalltalk
- Ada 95

Same Drawbacks as 4GL

Improves:
- Maintainability
- Reusability
- Modifiability

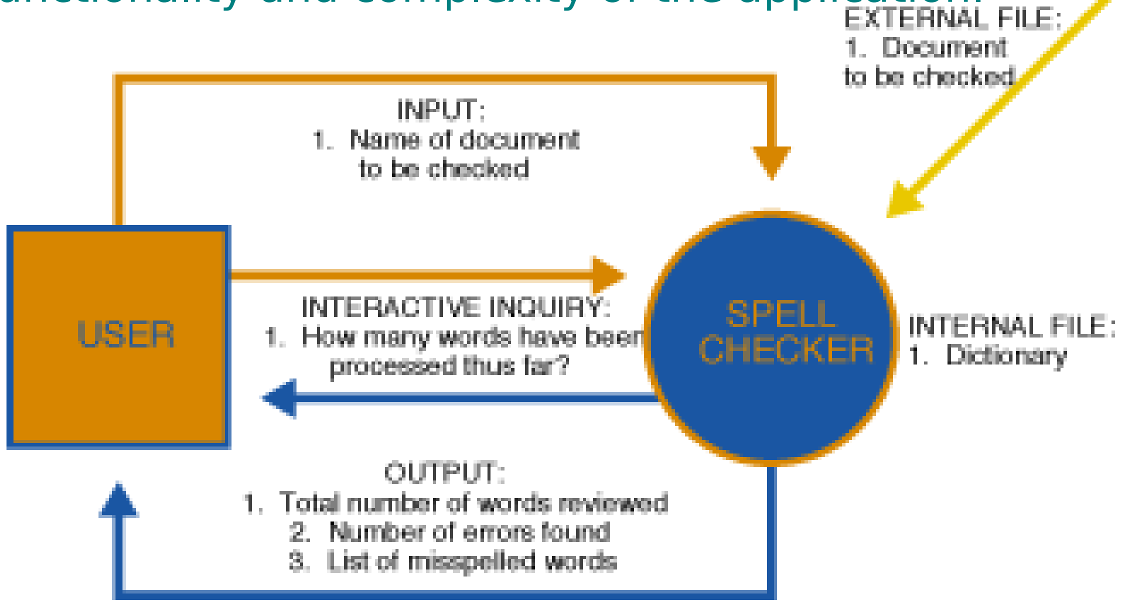
Overview of Software Sizing

- **Source Lines of Code (SLOC)**
  - Delivered Source Instructions (DSI)
  - Logical or Physical
  - Code Counters – consistent definition
  - All Executable Source Lines
    - Deliverable Job Control Statements
    - Data Declaration Statements
    - DATA TYPING and EQUIVALENCE statements
    - INPUT/OUTPUT format statements
- **Function Points (FPs)**
  - Measures 5 attributes (Inputs, Outputs, Interactive Inquires, External Files & Internal Files)
  - Adjusted or Unadjusted
  - International Function Point User’s Group (IFPUG) Definition or other
  - Counted manually – certified FP Counter
- **Object Points (Ops)**
  - Objects, Classes, POPs
- **Forms**
  - Screen/Window, Batch Process, Report, and Database Table

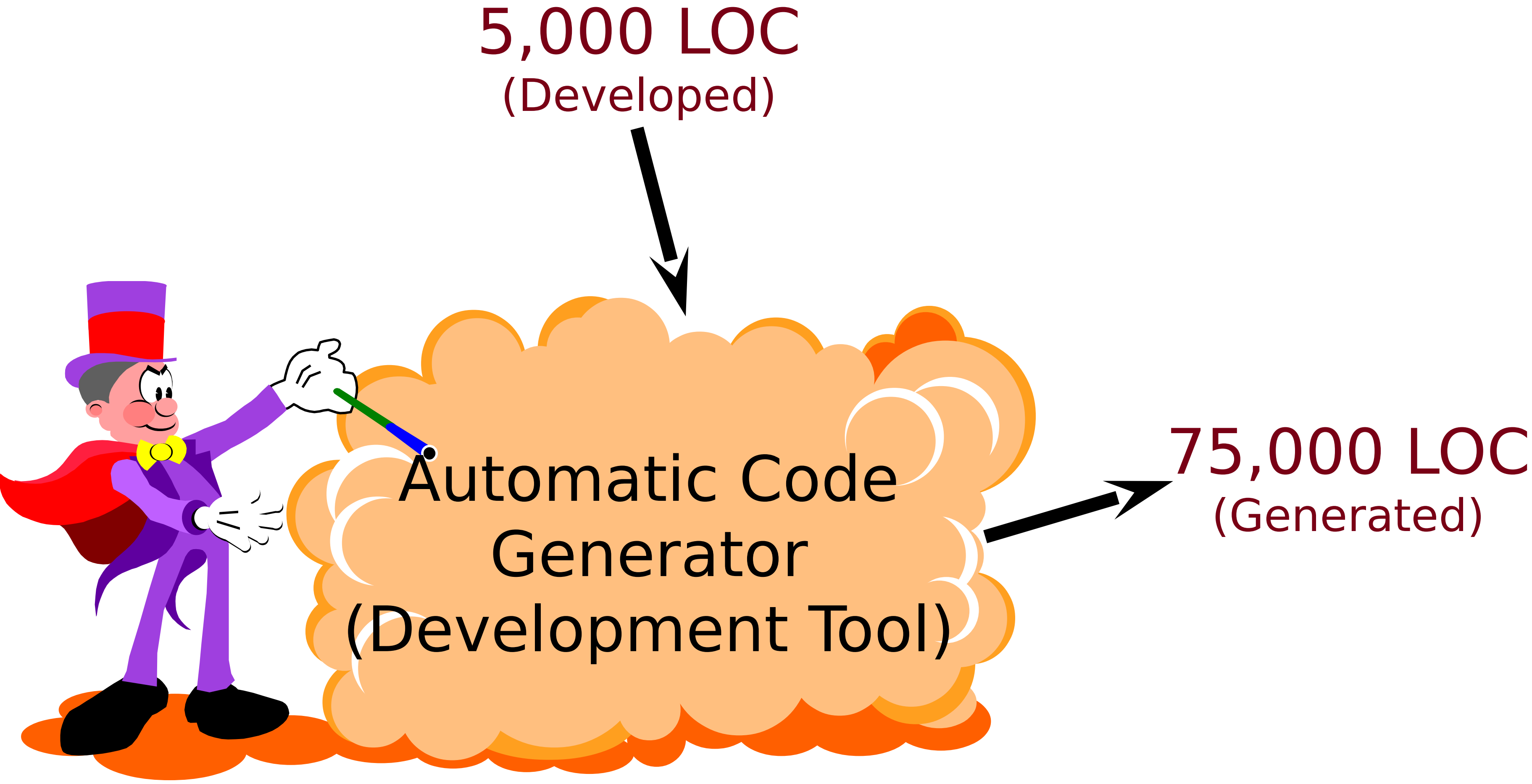
Overview of Software Function Points

FUNCTION POINTS

Provides a means to assess the size of a program in terms of its capability. The measurement requires the examination of five attributes of an application: its inputs, outputs, interactive inquiries, external files and internal files. For a simple spell checker, the number of such elements is seven, and each item needs to be weighted according to its individual complexity. The weighted sum of function points is then either increased or decreased to match the perceived intricacy of the overall program, as judged with 14 criteria. The final total will thus indicate the functionality and complexity of the application.


EXTERNAL FILE:
1. Document to be checked
INPUT:
1. Name of document to be checked
USER
INTERACTIVE INQUIRY:
1. How many words have been processed thus far?
SPELL CHECKER
INTERNAL FILE:
1. Dictionary
OUTPUT:
1. Total number of words reviewed
2. Number of errors found
3. List of misspelled words

Overview of Software Code Generators



5,000 LOC

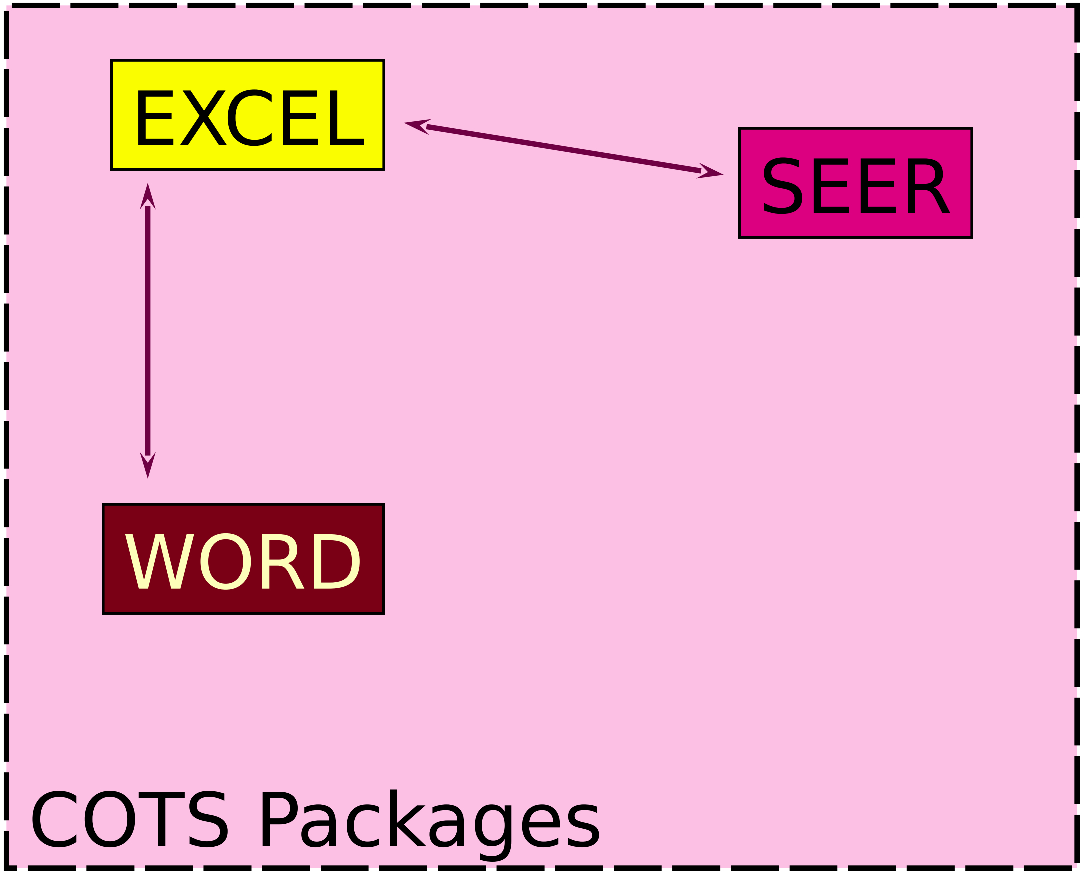
(Developed)

Automatic Code Generator (Development Tool)

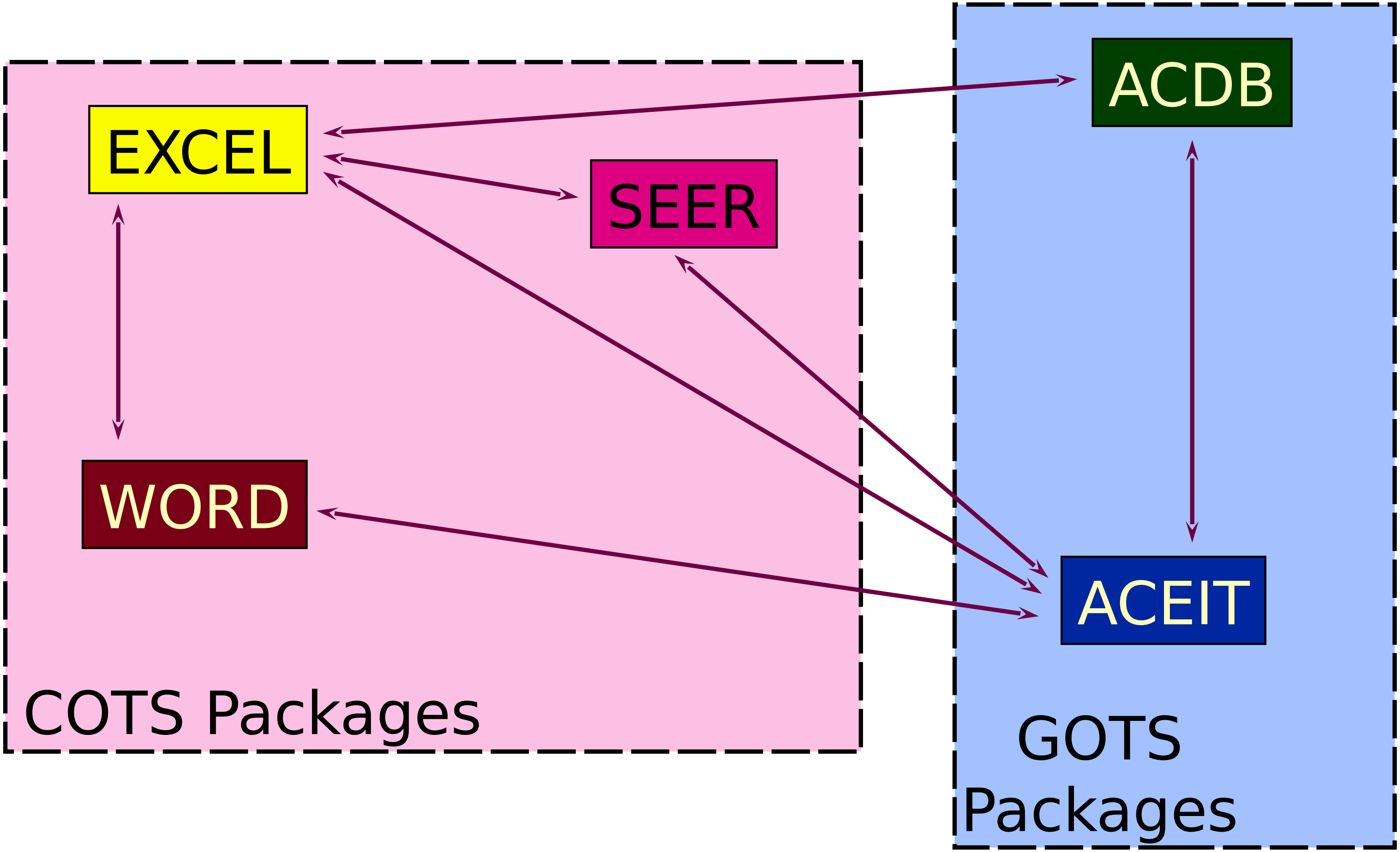
75,000 LOC

(Generated)

Overview of Software COTS/GOTS Integration



Overview of Software COTS/GOTS Integration



Ada Byron King Countess of Lovelace (1815-1852)

- Assistant to Charles Babbage for his “Analytical Engine”
- World’s First Programmer
- DoD’s way to blame a woman for all the problems

Overview of Software Ada

Why Ada ?

Develop a common single high order language for mission critical computer applications

- Real Time
- Modern Programming Techniques
- Large Scale Systems
- Maintainable

Overview of Software
Ada Features

- ANSI/ISO/IEC Standard (8652:1995)
  - Compiler Validated
  - Strict enforcement of standard
- Information Hiding (Encapsulation)
- Object Oriented Design
  - Modularity (Logical Structure)
  - Packaging
  - Exception Handling
  - Tasking
  - Generics
  - Abstract
- Strong Typed Language
- Legible Style

Overview of Software Ada 95

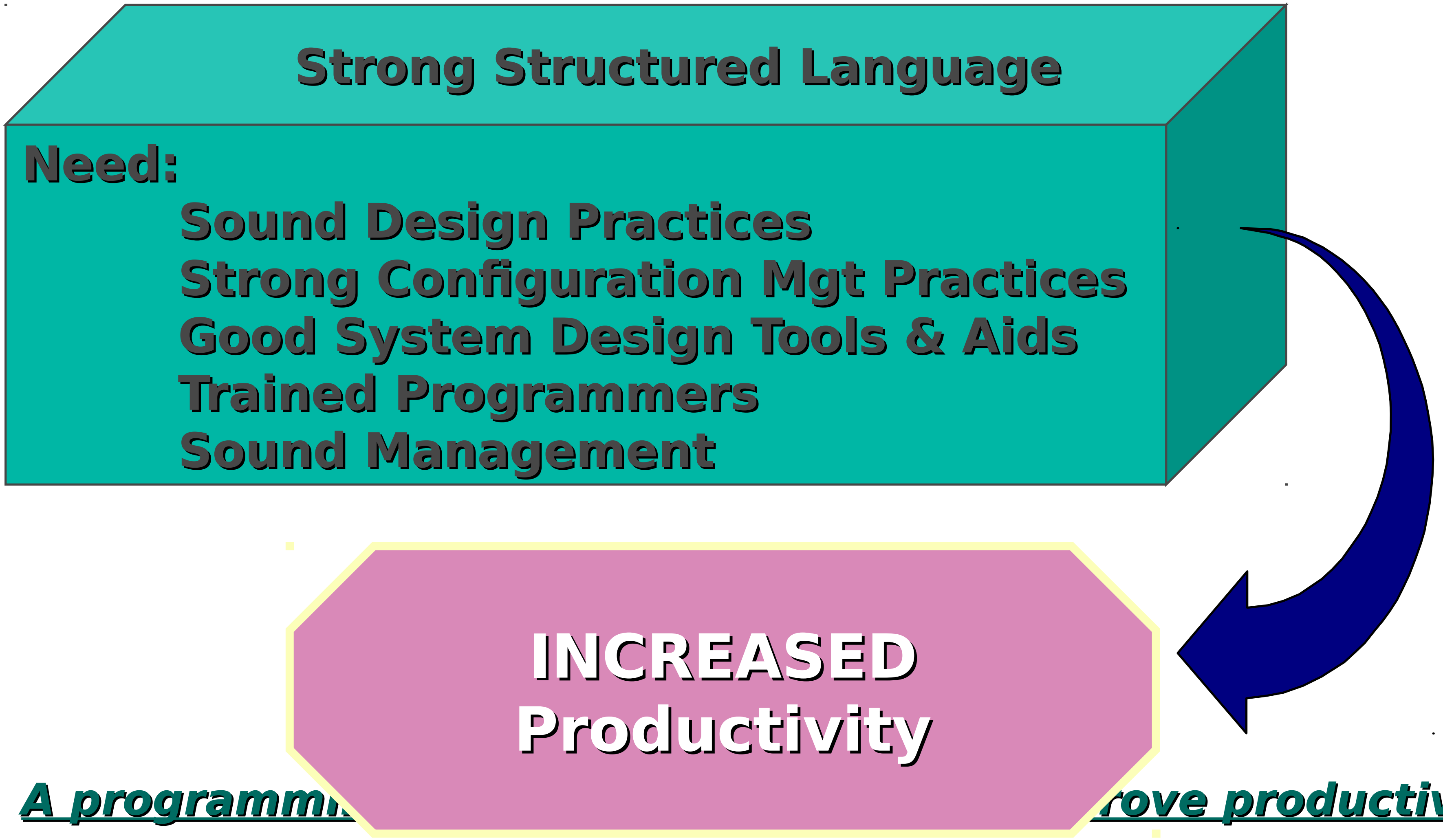
- Object Oriented Programming (OOP) support
- More efficient real time & parallel programming
- Upward Compatible
- International Character Sets
- Improved Generic Templates
- Faster compilation time of large systems
- Easier safety & security certification

Overview of Software DoD History

		1996
1966	Structured Programming	15% Use
1977	Structure Design	3% Use
	Structured Analysis	<1% Use
	Object Oriented Design	<1% Use

1965	Avg LOC/SM 55	With:	Assembly Language Batch Runs No tools
1985	Avg LOC/SM 90-95	With:	HOL Interaction Tools
1995	Avg LOC/SM 95-100	With:	HOL/Ada Object Oriented
Design			
			Tools
2000	Avg LOC/SM 95-120	With:	4GL & OO

Overview of Software Productivity Improvers



Strong Structured Language

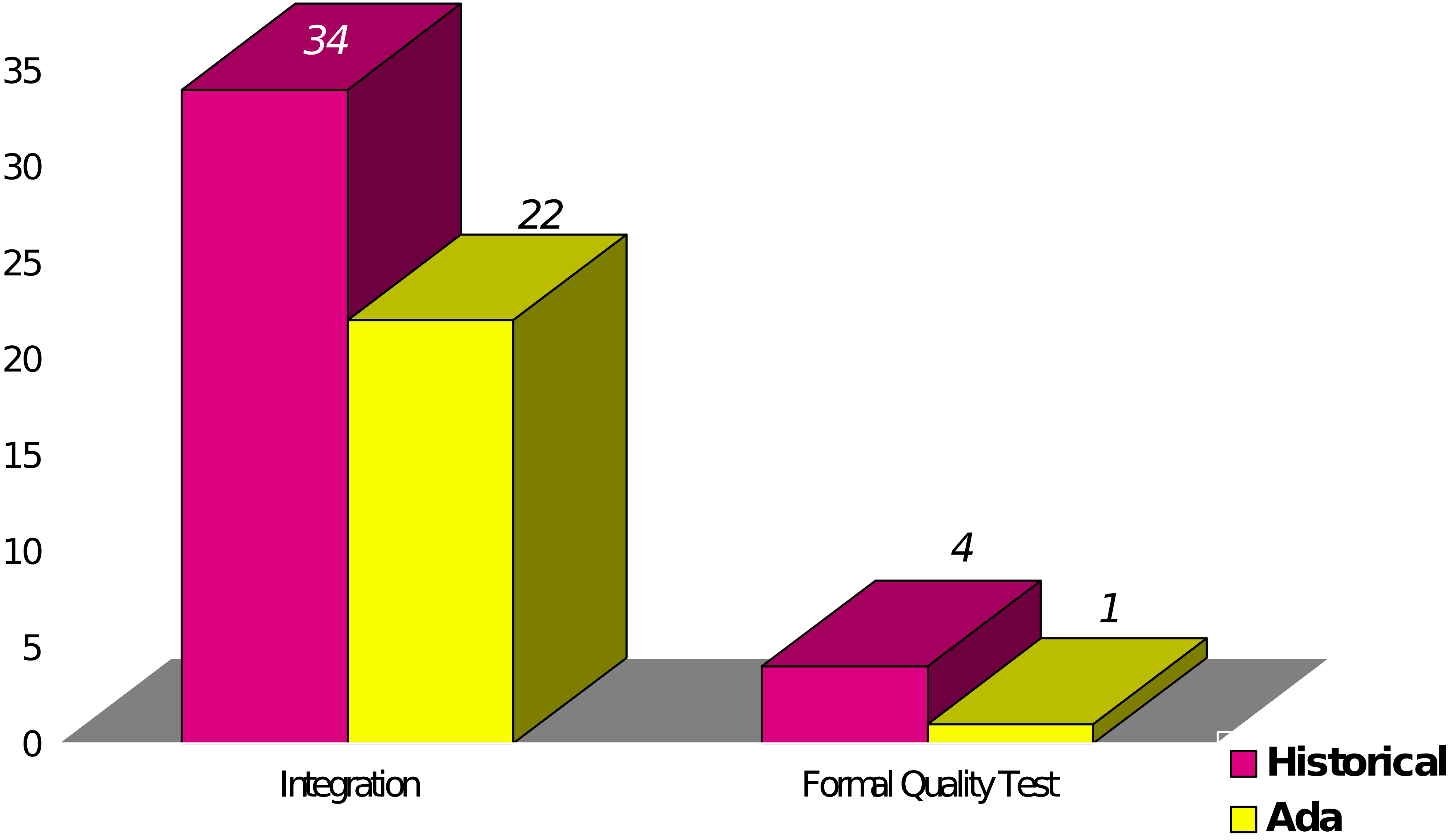
Need:

- Sound Design Practices
- Strong Configuration Mgt Practices
- Good System Design Tools & Aids
- Trained Programmers
- Sound Management

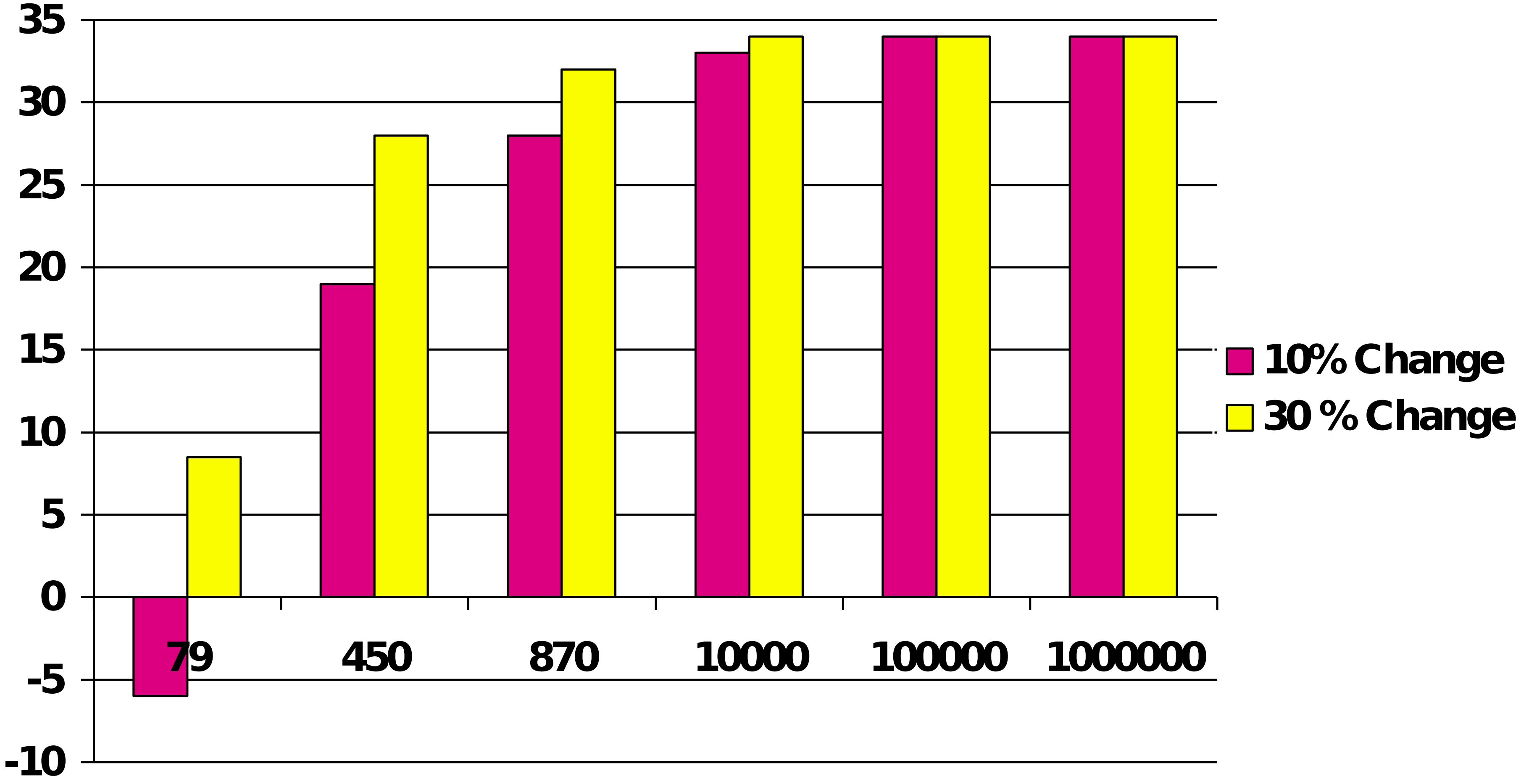
INCREASED Productivity

A programmi ***rove productiv***

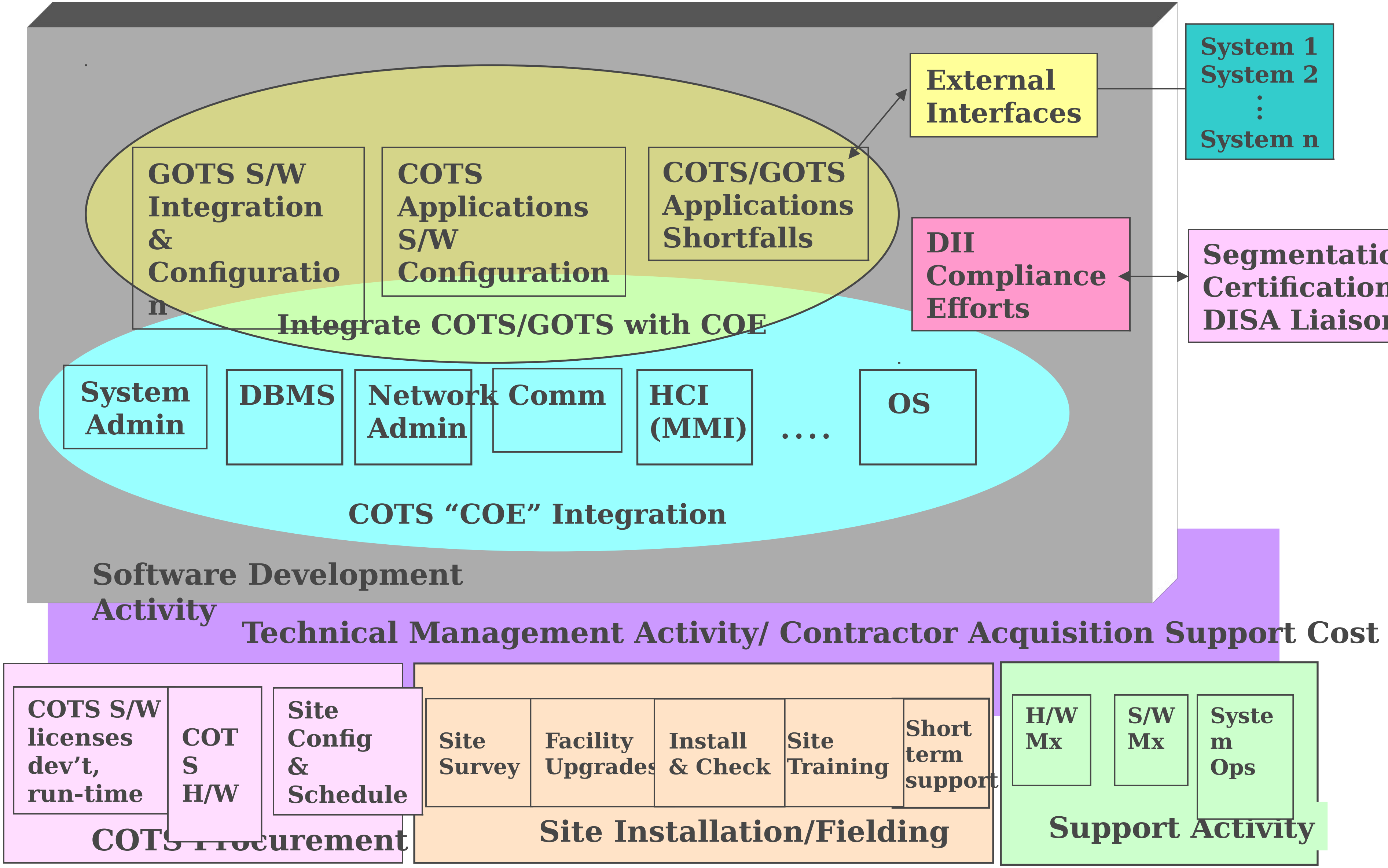
Error Rates
(per thousand Lines of Code)



Percent Savings in Maintenance
Ada vs. Other HOLs



Software Intensive Program Life Cycle Activities



GOTS S/W Integration & Configuratio n

COTS Applications S/W Configuration

COTS/GOTS Applications Shortfalls

External Interfaces

System 1
System 2
⋮
System n

DII Compliance Efforts

Segmentatic Certification DISA Liaisor

Integrate COTS/GOTS with COE

System Admin

DBMS

Network Admin

Comm

HCI (MMI)

....

OS

COTS “COE” Integration

Software Development Activity

Technical Management Activity/ Contractor Acquisition Support Cost

COTS S/W licenses dev't, run-time

COT S H/W

Site Config & Schedule

COTS Procurement

Site Survey

Facility Upgrades

Install & Check

Site Training

Short term support

Site Installation/Fielding

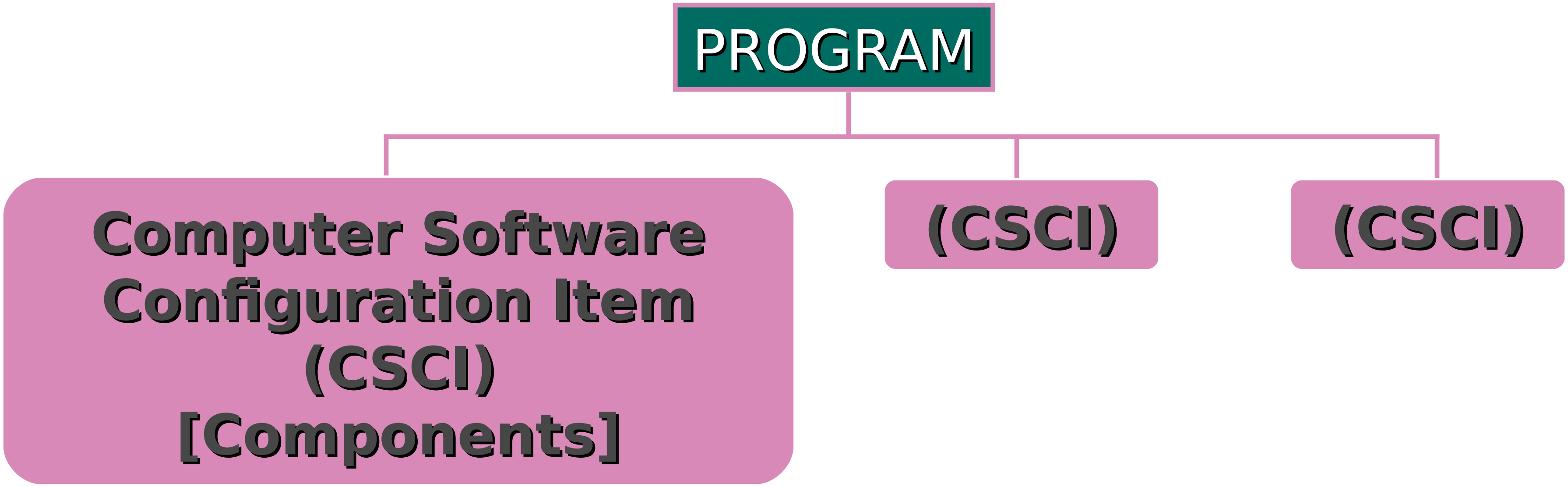
H/W Mx

S/W Mx

Syste m Ops

Support Activity

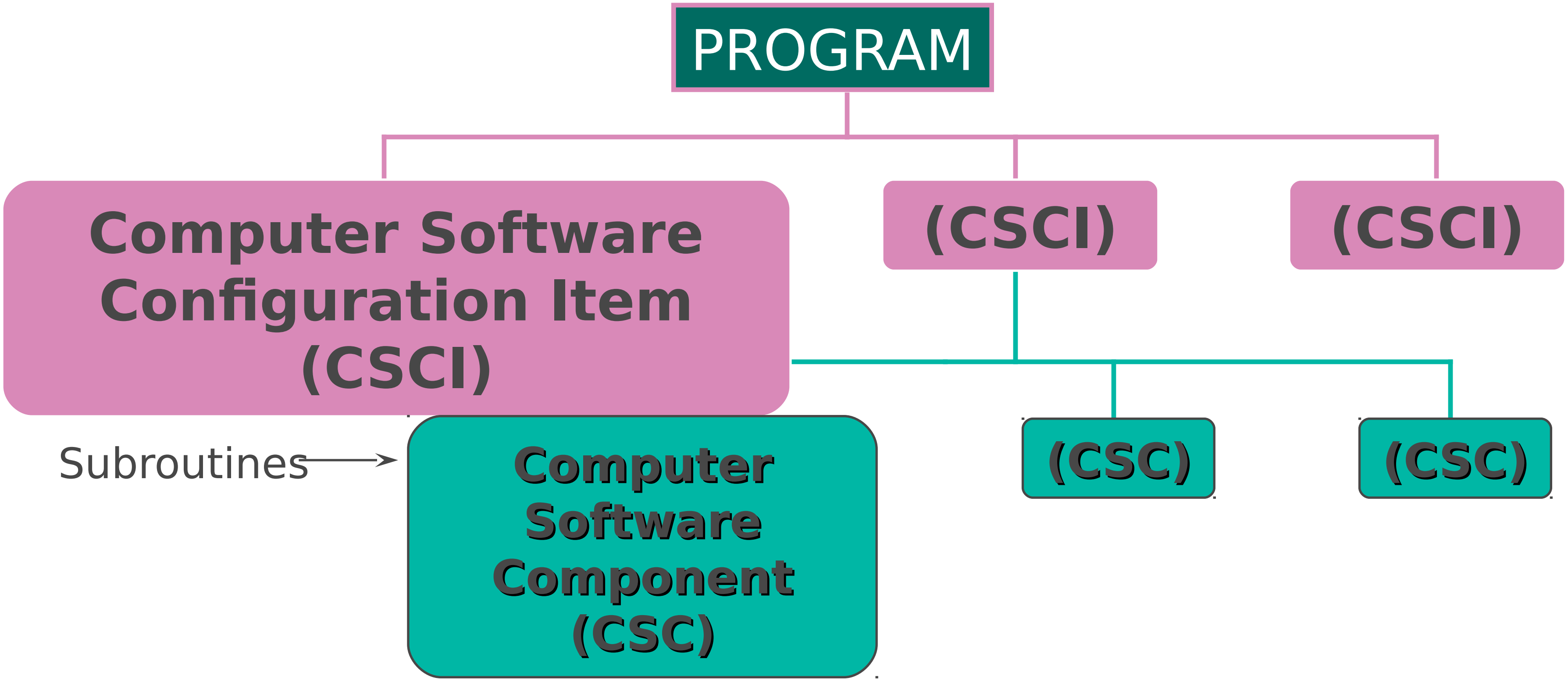
Overview of Software Software Structural Breakdown



Typical Sizes

CSCI 100,000 - 150,000 LOC

Overview of Software
Software Structural Breakdown

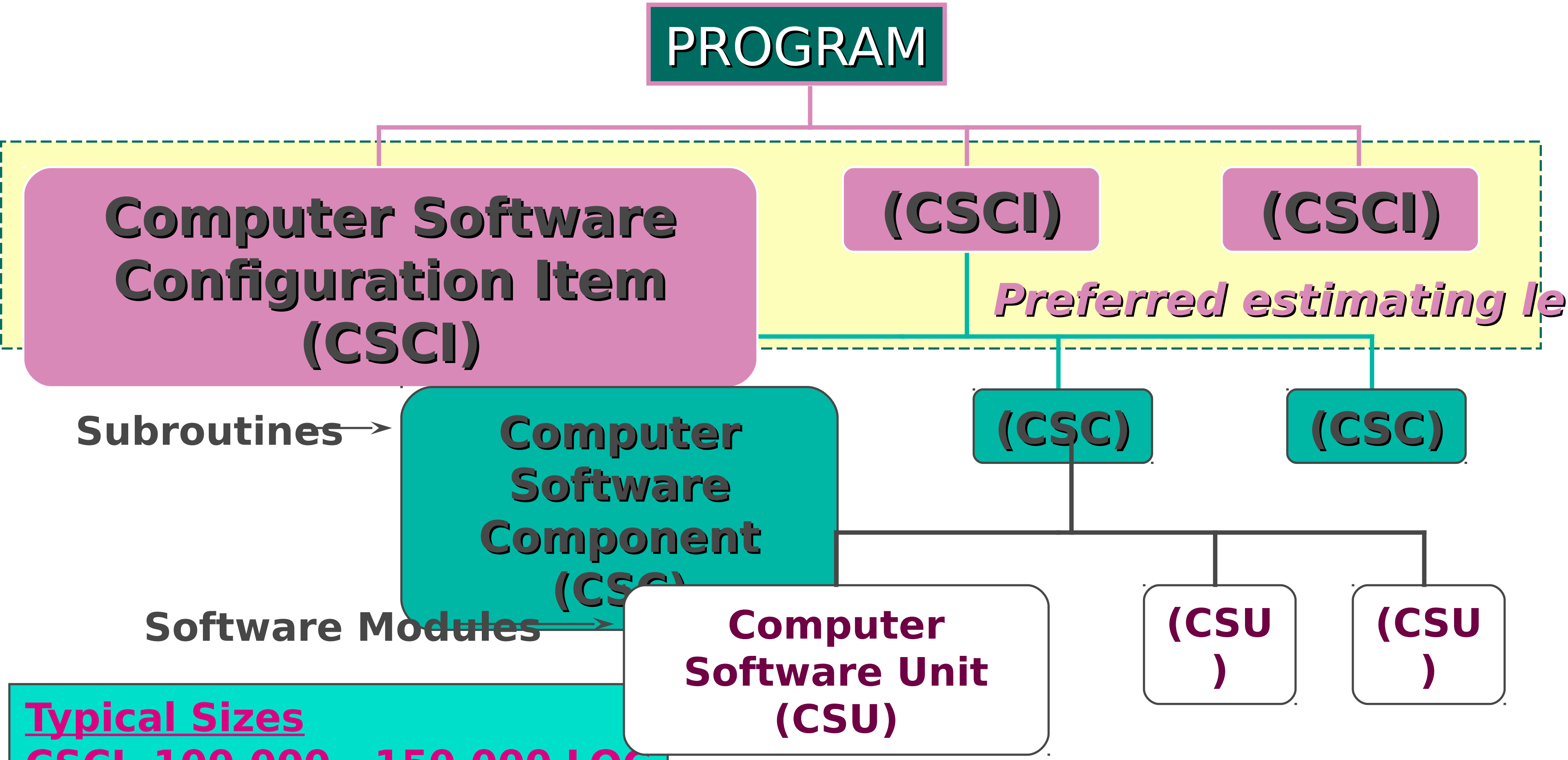

PROGRAM
Computer Software Configuration Item (CSCI)
(CSCI)
(CSCI)
Subroutines
Computer Software Component (CSC)
(CSC)
(CSC)

Typical Sizes

CSCI 100,000 - 150,000 LOC

CSC 1,000 - 5,000 LOC

Overview of Software
Software Structural Breakdown



PROGRAM

Computer Software Configuration Item (CSCI)

(CSCI)

(CSCI)

Preferred estimating le

Subroutines

Computer Software Component (CSC)

(CSC)

(CSC)

Software Modules

Computer Software Unit (CSU)

(CSU)

(CSU)

Typical Sizes

CSCI 100,000 - 150,000 LOC

CSC 1,000 - 5,000 LOC

CSU 100 - 500 LOC

Overview of Software
X X
MIL-STD Handbook 881B

Prime Mission Product

- **Software - Developed**
  - **CSCI 1 (Component 1)**
  - **CSCI 2 (Component 2)**
  - **COTS Integration**
  - **Software Integration**

Software Cost Estimating Contents

- Overview of Hardware & Software
- ✓ ***Steps of a Software Cost Estimate***
- SEER SEM
- Specific to ESC
- Common Mistakes
- Current Issues & Conclusions

Steps of a Software Estimate

- Understand the Program & Scope
- Select Methodology & Collect Technical Information
- Analyze Technical Information
- Reconcile and Coordinate
- Generate an Estimate
- Fill in Missing Pieces
- Perform Confidence Checks
- Present Information

Understand the Program & Scope

- Grasp an understanding of the TOTAL Program
  - Read the PMD
  - Review any previous estimates
  - Program Managers Overview

Understand the Program & Scope

- Grasp an understanding of the TOTAL Program
- Find out the top level specifics of the software
    - What are the main functions ?
    - How does everything fit together ?
    - What is the scope of the work? (Software Requirements)

Understand the Program & Scope

- Grasp an understanding of the TOTAL Program
- Find out the top level specifics of the software
- Find out developmental process
  - What is the developmental approach ?
  - What mil-standards are being used?

Software Standards

DoD-STD 2167	1984-1987
DoD-STD 2167A	1988-1994

Dod Standards designed to help standardize software development

On June 29, 1994, Secretary of Defense William J. Perry issued memorandum - "A New Way of Doing Business" MIL-STDs are no longer mandated - need waiver to use a MIL-STD

MIL-STD 478	1994-1996

Navy & Air Force issued blanket waivers allowing use of this MIL-STD

J-STD 016
ISO 12207

US is combining these two standards to form US 12207

The Standards applied determine:

- the amount and level of documentation
- the level of reporting
- the number of reviews and when they occur
- the level of testing and quality assurance

Understand the Program & Scope

- Grasp an understanding of the TOTAL Program
- Find out the top level specifics of the software
- Find out developmental process
- Find out all contractual information
  - Are there any subcontractors ?
    - Who are they?
    - What are they doing?
    - What percent complete are they?
    - Is there any actual information available?
    - Have they had any pitfalls? What?

Understand the Program & Scope

- Grasp an understanding of the TOTAL Program
- Find out the top level specifics of the software
- Find out developmental process
- Find out all contractual information
- Purpose of the estimate
  - Aid the SPO in defining the software definition
  - Determines the level of detail
  - Assists in determining any alternatives or modifications

Select Methodology

Primary:

- Software Estimating Models
- Analogous Programs
- Cost Estimating Relationships (CERs)

Confidence Checks/ Secondary:

- Software Estimating Models
- ESC Factors

Select Methodology

Methodologies NOT recommended for estimating Software Development

CPR Analysis

Manloading (Grass Roots)

Select Methodology
Software Estimating Models

- Cost Estimating Models
  - SEER SEM
  - PRICE S
  - COCOMO ‘83
  - SLIM
  - REVIC
  - SASET
  - COCOMO ‘02

Select Methodology Analogous Programs

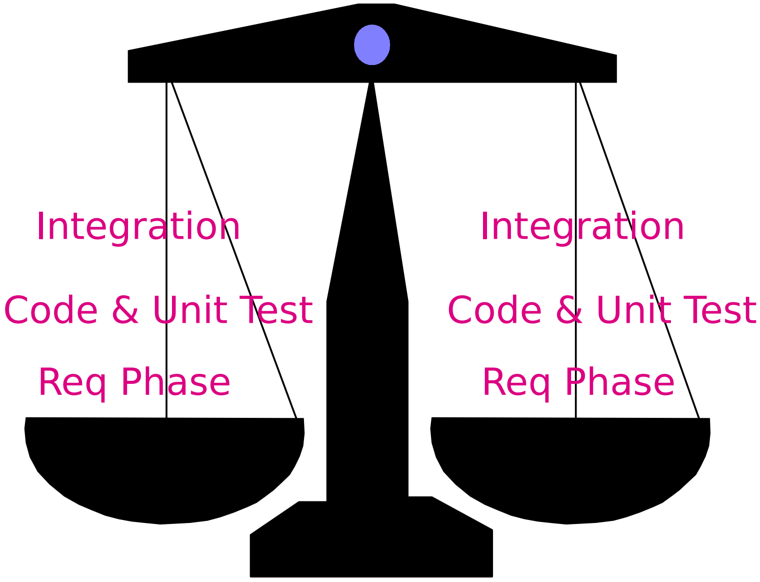
- Similar Technical Information
- Same Type of Team
- Same Cost Driving Parameters
- Available Data
  - ➢ Data Collection
    - – How?
    - – When?

FRONT-END ESTIMATING

Select Methodology
Calculating Productivities

Productivity = Lines of Code/Staff Month
= LOC/SM

Know what is included in the SM
- What phases of development
- What people



Best not to use $/LOC
- Staff Rate dependen
- Includes inflation

Select Methodology
ESC Factors

ESC Pamphlet 173-2

- Section A: Acquisition Factors & CER
- Section B: Software Factors
- Section C: Beta Curve Distributions

Collect Technical Information
Determine Software POCs

- Want the senior most knowledgeable person

  From at least one of the following:

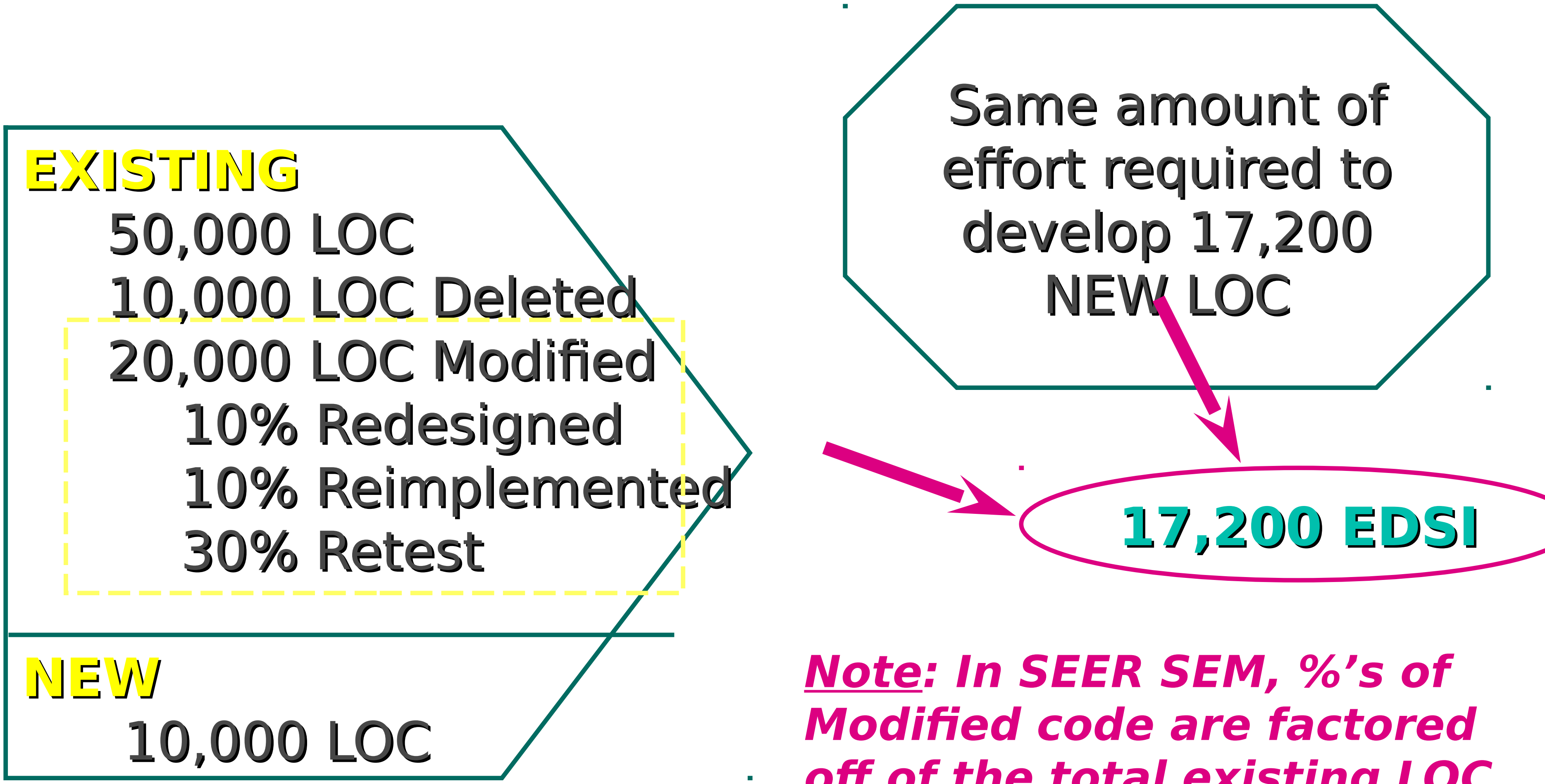
  - Program Office
  - Mitre
  - Software Development Contractor
  - IV & V Contractor

- Keep good relations (communication) on-going

Collect Technical Information
Input Parameters

- Take input sheets personally to POCs
  - Go through an example CSCI explaining any parameters they have questions about
- POCs fill out independent of each other
- Need parameter sheet for each CSCI of each build (block or phase) or subsystem
- Set a specific deadline for the input sheets to be completed by (rule of thumb - 1 week)
- Be sure to collect the most likely range - not the chance in a million

Equivalent Deliverable Source Instructions (EDSI) - Effective Size



EXISTING

50,000 LOC

10,000 LOC Deleted

20,000 LOC Modified

10% Redesigned

10% Reimplemented

30% Retest

NEW

10,000 LOC

Same amount of effort required to develop 17,200 NEW LOC

17,200 EDSI

***Note**: In SEER SEM, %’s of Modified code are factored off of the total existing LOC.*

Collect Technical Information EDSI Equations

ADJUSTMENT FACTOR EQUATION:

ADJ = [(%Redesign) * D] + [(%Reimplementation * I] + [(%Re

Collect Technical Information
EDSI Equations

ADJUSTMENT FACTOR EQUATION:

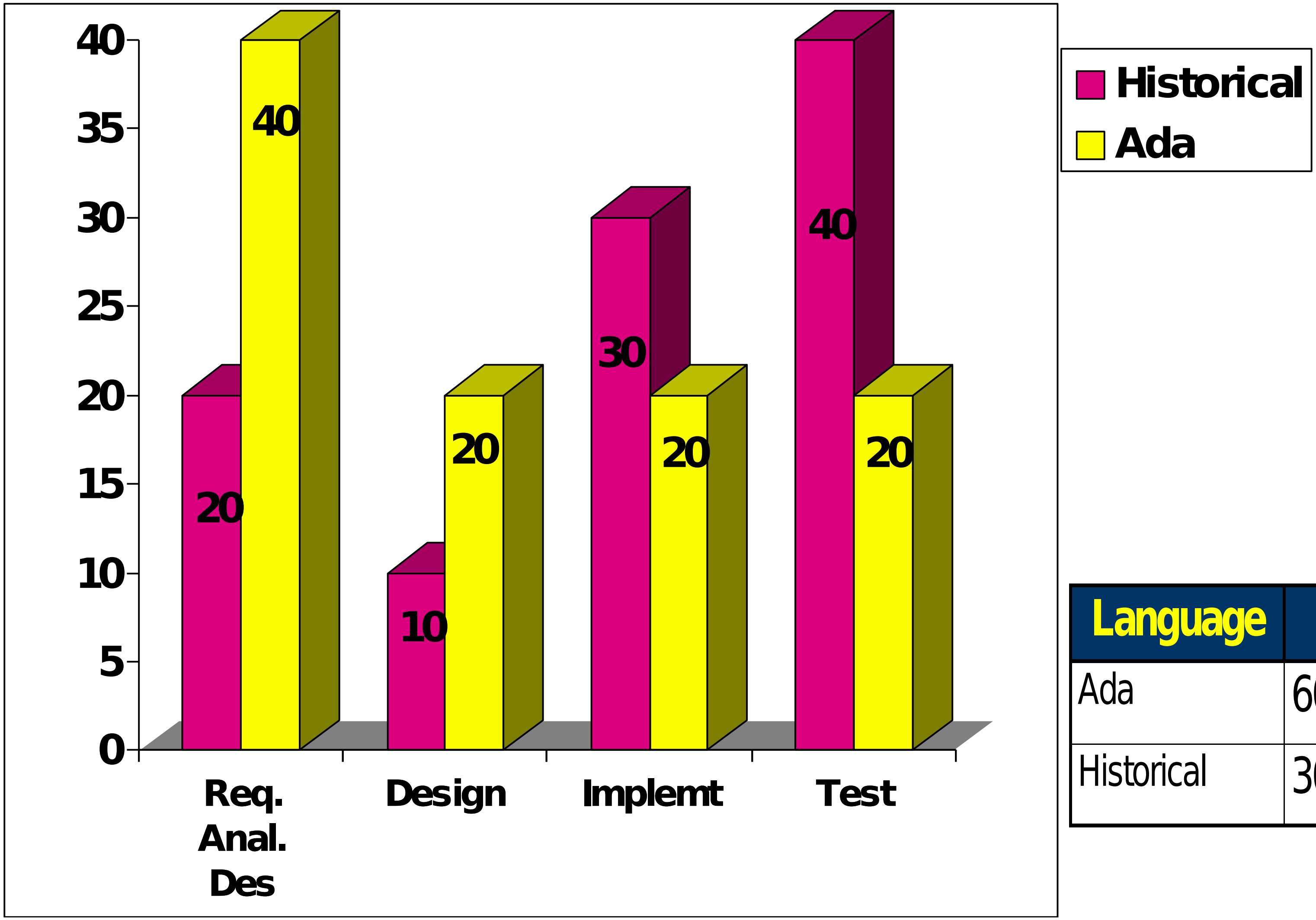
ADJ = [(%Redesign) * D] + [(%Reimplementation * I] + [(%Re

EFFECTIVE SIZE EQUATIONS:

EDSI = [(Existing LOC - Deleted LOC) * ADJ] + New LOC

Model	D	I	T
SEER SEM	40%	25%	35%
COCOMO	40%	30%	30%
COCOMO 2	40%	30%	30%

Historical Life-Cycle Phases



Language	D	I	T
Ada	60%	20%	20%
Historical	30%	30%	40%

Information SEER SEM EDSI Calculation

ADJ = [(%Redesign) * .4] + [(%Reimp * .25] + [(%Retest) * .35]

EDSI = [(Existing LOC - Deleted LOC) * ADJ] + New LOC

Example:	110,000 LOC Existing	30% Redesign
	70,000 LOC Deleted	35% Reimplementation
	10,000 LOC New	50% Retest

Information
SEER SEM EDSI Calculation

ADJ = [(%Redesign) * .4] + [(%Reimp * .25] + [(%Retest) * .35

EDSI = [(Existing LOC - Deleted LOC) * ADJ] + New LOC

Example: 110,000 LOC Existing 30% Redesign
70,000 LOC Deleted 35% Reimplementation
10,000 LOC New 50% Retest

ADJ = (.3 * .4) + (.35 * .25) + (.5 * .35)
= .12 + .0875 + .175
= .3825

EDSI = ((110,000 - 70,000) * .3825) + 10,000
= (40,000 * .3825) + 10,000
= 15,300 + 10,000
= 25,300

Collect Technical Information ReUse

- Libraries
- Incremental Development
- Modifications
- Common Code
  - Internal
  - External

Collect Technical Information Input Parameters

- Personnel Capabilities & Experience
    - ➢ Rate the team as a whole (not one hot-shot)
- Modern Programming Practices & Tools
    - ➢ Must use them not just have them
- Inputs are by CSCI not by program

Analyze Technical Inputs

- Check to see what the differences are between CSCIs
  - Do they all look the same ?
- Look for extreme ratings
  - Very High and above
  - Very Low and below
- Anything that goes against your gut feeling be sure to get supporting rationale
- Notice wide ranges from the least to most

Analyze Technical Inputs By Parameter

- Size -- Range 5,000 - 150,000 Total LOC
- Personnel
  - Typically not higher than Nominal +
- Programming Practices & Tool
  - Should not be rated high just because using Ada
- Language Type
  - Not all Ada Programs are rated at High
- Target Environment
  - Special Display Requirements, Time Constraints, Real Time Code, Security Requirements (Generally, not > Nom for more than 2 CSCIs per program)

Analyze Technical Inputs
Input Parameters

With today’s technology, the following are typically no longer cost drivers

- Memory Constraints
- Hardware Volatility
- Compilers Volatility
- Ada Experience
- Database Size

Analyze Technical Inputs Odds & Ends

- Software to Software Integration
  - Get an integration schematic and verify
- Software to Hardware Integration
  - Not usually included in software costs
  - Not separated from S/W to S/W integration on SEER outputs
- Staff Month
  - Use actuals if on contract or current rate
  - Be sure to include any subcontractor loadings
  - Recommendation: Do everything in staff months & let ACE calculate your dollar values

Analyze Technical Inputs
Capability Maturity Model (CMM)

	LEVEL	TRAIT	PROBLEM	COMMENT
1	INITIAL	Ad Hoc	Planning	Lack of management awareness
2	REPEATABLE	Intuitive	Training	Needed to obtain experts
3	DEFINED	Process Defined	Measurement	Need data to assess technology
4	MANAGED	Measured Process	Technology	How to best use technology to close feedback loop
5	OPTIMIZING	Process Feedback	Automation	What can we automate

Page 71

Analyze Technical Inputs Final Comments

- Go back to the engineers as many times as need be
- Do not be afraid to ask lots of questions
- Always get supporting rationale to back up whatever information you are given

Reconcile and Coordinate

- Reconcile all parameter inputs
  - Differences
    - Why?
    - Who do you believe?
    - If needed, get all parties together to resolve any open issues.
- Coordinate Everything
  - Be sure the Program Manager is willing to sign off on all the inputs
  - Have an informal briefing with all key players to discuss the final technical baseline you have established.

Generate an Estimate

- Push the button and away you go..........

Dollarize your estimate

Hints:

- The upfront SEER questions do not matter if you are going to do inputs by individual parameters
- If alot of the same parameters for each CSCI, then create a default set.

Generate an Estimate
Labor Rates

- Contractor Rate
  - Contractor Provided
  - CPR Analysis
- IDIQ Model (Automated BLS)
- Industry Average

Generate an Estimate Labor Rate (Cont.)

e sure to Include:

- Proper Labor Categories
  - Systems Engineer
  - Systems Analyst
  - Software Engineer
  - Programmer
- Appropriate Loadings
  - Overheads
  - G & A
  - Profit/Fee
- Correct # of Labor Hours per month

Labor Rate Calculation Example

Prime Software Contractor:	S/W Subcontractor:
$30 per staff hour	$23 per staff hour
158 hrs in a staff month	152 hrs in a staff mont
115% Overhead	110% Overhead
12% G & A	15% G & A
10% Profit	9 % Profit

Labor Rate Calculation Example

Prime Software Contractor:

$30 per staff hour
158 hrs in a staff month
115% Overhead
12% G & A
10% Profit

SM Rate = 30 * 158
= $4740

OH: 4740 * 2.15 = $10,191
G&A: 10191 * 1.12 = $11,413.92
Profit: 11413.92 * 1.1 = **$12,555.31**

S/W Subcontractor:

$23 per staff hour
152 hrs in a staff mont
110% Overhead
15% G & A
9 % Profit

SM Rate = 23 * 152
= $3496

OH: 3496 * 2.1 = $7,341.60
G&A: 7341.6 * 1.15 = $8,442.84
Profit: 8442.84 * 1.09 = **$9,202.70**

With Prime Loadings:
9202.7 * 1.12 * 1.1 =

Generate an Estimate Time Phasing

- Actual History - if on contract
- Beta Curves
  - Analogous Program
  - Prior Build
  - ESC History for Software Development (ESCP 173-2C)

Fill in Missing Pieces Risk

- Where is it? (Add risk Discretely)
  - Areas of discrepancy
  - Parameters with large ranges
- Is it already included?
  - Risk should not be included in initial ratings
  - Ask engineers specifically to identify areas
  - Not chance in a million

Fill in Missing Pieces
Risk

- Common areas of risk
  - Lines of Code
  - People
  - Changing Requirements
  - Security
  - Reliability

Fill in Missing Pieces
Firmware

- Software that is permanently burned onto hardware
- Managed like software
- Estimate like software
- Usually has a very high reliability

Fill in Missing Pieces Maintenance

- Maintenance - modifying existing operational software while leaving its primary functions in tact.

Fill in Missing Pieces Maintenance

- Maintenance - modifying existing operational software while leaving its primary functions in tact. Two main categories:
  - Software Updates - Changed Functional Specification
  - Software Repair - Leaves Functional Spec intact
    - Corrective maintenance (of processing, performance, or implementation failures)
    - Adaptive maintenance (to changes in the processing or data environment)
    - Perfective maintenance (for enhancing performance or maintainability)

Fill in Missing Pieces
Maintenance

- At time of software delivery, maintenance begins
- If software is being developed incrementally, be sure to include any maintenance needed on previous blocks

Software Maintenance Methodologies

“Card Ratio” Method

Ratio Factor = # of LOC one person can maintain per month

Determined by:

- Reliability built into the code
- Application Type - (Weapon System or MIS)
- Language & Size
- People (Developers & Maintainers)

Range: (25,000 - 200,000)

Average: 150,000

$$\frac{\text{Total Delivered LOC}}{\text{Ratio Factor}} = \text{\#People per mth of Maint}$$

Software Maintenance Methodologies

Maintenance/ Development Cost Ratio

Maint Cost = Maint/Devl Ratio * Development Costs

Maint/Devl Ratio range : 0.67 - 4.5

Average : 1.5
(Corresponding to a 60% Maintenance,
40% Development Life-Cycle)

Software Maintenance Methodologies

Annual Change Traffic (ACT) - that fraction of the software product’s LOC which undergo change during a “typical” year - either addition or modification.

ACT range : 1.0% - 15%

Average : 5%

LOC Mod per Year = Total LOC * ACT

$$\text{Maint (SM per YR)} = \frac{\text{LOC Mod per Yr}}{\text{Productivity (LOC/SM)}}$$

Fill in Missing Pieces
Warranty

- Warranty - Agreement to fix bugs within a set period of time after software delivery
    - Use is not recommended for software

Fill in Missing Pieces

- Independent Verification and Validation (IV&V)
  - If applicable include in total program estimate
  - Range: 5% - 40% of developed software costs
- Commercial Off-The-Shelf (COTS) Software:
  - Not included in the developed software line
  - Be sure to capture all integration efforts

Perform Confidence Checks

- Secondary Model
  - COCOMO
    - Parameters can easily be translated from SEER inputs
    - Only 2 additional inputs: Data and Schedule
- Lines of Code per Staff Month(LOC/SM)

  Be sure comparing Apples to Apples
  - Analogous Programs
  - ESC History
  - ESC Software Database
    - Includes: Requirements, Development, S/W to S/W Integration
    - Excludes: S/W to H/W Integration, IV&V, Maintenance

Perform Confidence Checks COCOMO

- Comparing Estimates to SEER Estimates
- Parameter Translation
  - COSTAR - allows you to do direct SEER translation
- Additional Add-Ons
  - Security
  - Requirements Change Volatility
- Compare at the EMD level
  - COCOMO does not include upfront requirements phase
  - COCOMO does not include CSCI to CSCI integration

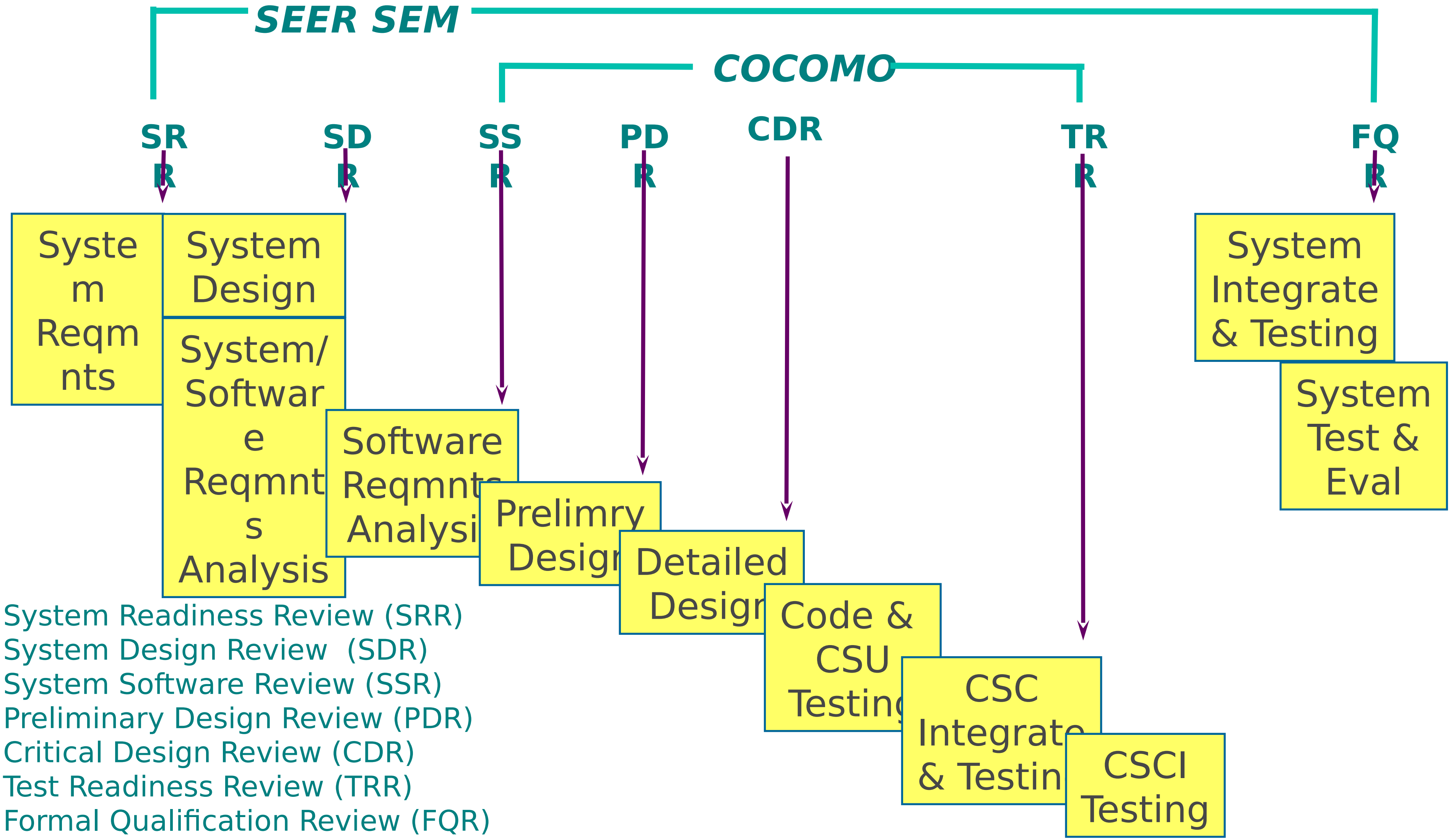
SEER to COCOMO Translation

SEER	COCOMO
Complexity	
Analyst Capabilities & Experience	ACAP
Analyst Application Experience	AEXP
Programmer Capabilities	PCAP
Programmer Language Experience	LEXP
Host Development System Experience	VEXP
Target System Experience	VEXP
Modern Development Practices Use	MODP
Automated Tool Use	TOOL
Logon thru Hardcopy Turnaround Time	TURN
Host Development System Volatility	VIRT

SEER to COCOMO Translation

SEER	COCOMO
Requirements Volatility (Change)	(Add-On)
Specification Level-Reliability	RELY
Test (Verification/Validation) Level	RELY
Quality Assurance Level	RELY
Language	(Database)
Application Class Complexity	CPLX
Memory Constraints	STOR
Time Constraints	TIME
Target System Volatility	VIRT
Security Requirements	(Add-On)
NO EQUIVALENT PARAMETER	DATA
NO EQUIVALENT PARAMETER	SCHED

Checks
Lifecycle Comparison (DoD STD 2167A)



SEER SEM

COCOMO

SR
R

SD
R

SS
R

PD
R

CDR

TR
R

FQ
R

Syste
m
Reqm
nts

System
Design

System/
Softwar
e
Reqmnt
s
Analysis

Software
Reqmnts
Analysi

Prelimry
Design

Detailed
Design

Code &
CSU
Testing

CSC
Integrate
& Testin

CSCI
Testing

System
Integrate
& Testing

System
Test &
Eval

System Readiness Review (SRR)
System Design Review (SDR)
System Software Review (SSR)
Preliminary Design Review (PDR)
Critical Design Review (CDR)
Test Readiness Review (TRR)
Formal Qualification Review (FQR)

Present Information

- Charts
  - LOC (EDSI, New, Existing)
  - Staff Months
  - LOC/SM
  - By CSCI
  - By Development Phase (Req, EMD, Integ)
- Documentation
  - All Supporting Rationale
  - Summary Spreadsheets (LOC, Effort, Parameters)
  - Risk estimate and Rationale
  - Model descriptions
  - Model runs

Software Cost Estimating Contents

- Overview of Hardware & Software
- Steps of a Software Cost Estimate
- ✓ ***SEER SEM***
- Specific to ESC
- Common Mistakes
- Current Issues & Conclusions

SEER SEM

- Size Parameter
  - Lines of Code (LOC)
  - Function Points
- 34 Technical Parameters
  - Complexity
  - Personnel Capabilities & Experience
  - Development Support Environment
  - Product Reusability Requirements
  - Development Environment Complexity
  - Target Environment

SEER SEM (Cont..)

- Software Requirements Analysis
- Software to Software Integration
- Software to Hardware Integration

SEER SEM Validation

	ACTUALS		SEER SEM		
Program	SM	LOC/SM	SM	LOC/SM	%Diff
Project 1 LOC = 100K Lang = Ada	642	148	596	159	7%
Project 2 LOC = 262K Lang = Fortran	5,690	46	5,630	47	1%
Project 3 LOC = 102K Lang = Fortran	1,222	84	1,220	84	0%
Project 4 LOC = 2,060K Lang = Jovial	27,443	75	25,630	80	7%

Software Cost Estimating Contents

- Overview of Hardware & Software
- Steps of a Software Cost Estimate
- SEER SEM
- ✓ ***Specific to ESC***
- Common Mistakes
- Current Issues & Conclusions

Application Software Definitions

- Weapon Systems:
  - Usually systems that are time critical in nature
  - Network monitoring, network control and switching, sensor control, signal/telemetry processing, message processing, data reduction/analysis, mission control, command processing, mission planning, message switching
- Non-Weapon Systems:

  MIS (Management Information System)/ AIS (Automated Information System)
  - Resource estimation, project planning, accounting, configuration management, performance monitoring, decision analysis

Support Software Definitions

- Simulation
  - Environment simulator, system simulation, emulation
- S/W Development Tools
  - Compiler, linker/loader, debugger, editor, assembler, requirements analysis, design tool aids, code generator, programming aids, report generator, code auditor
- Test Software
  - Test case generation, test case data recording, test case data reduction/analysis, test driver
- Training Software
  - Computer Aided Instruction (CAI), simulator, scenario generator
- Utilities
  - Media Conversion, sort/merge, format translation, math routines, plotting routines, input/output drivers, miscellaneous routines

Trends in Software Development

- Modifications/Enhancements rather than new systems
- Evolutions
- COTS Emphasis
- Upgrade or fix Hardware First

Software Cost Estimating Contents

- Overview of Hardware & Software
- Steps of a Software Cost Estimate
- SEER SEM
- Specific to ESC

✓ ***Common Mistakes***

- Current Issues & Conclusions

Common Mistakes

- LOC/SM
  - Be sure comparing apples to apples
  - Go with actuals rather than wishful thinking
- Parameters
  - By CSCI
  - Security
  - Existing LOC
  - Hardware Integration
- Don’t let engineers tell you how to estimate
- Talk to the right functional specialists

Software Cost Estimating Contents

- Overview of Hardware & Software
- Steps of a Software Cost Estimate
- SEER SEM
- Specific to ESC
- Common Mistakes
- ✓ ***Current Issues & Conclusions***

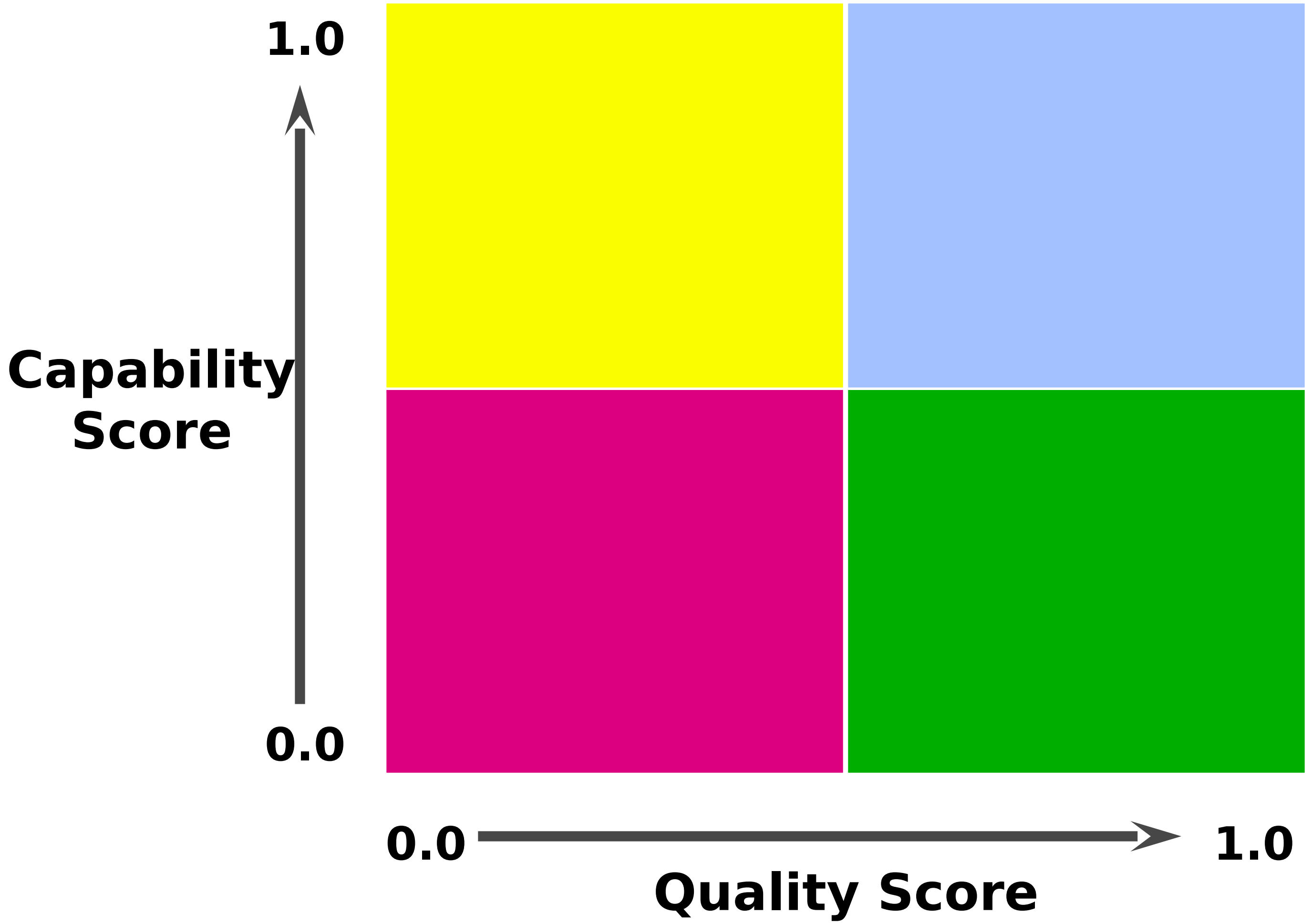
Current Issues & Conclusions Emphasis Change

Quality vs. Capability

Meeting the User's Needs

What can I get for my money?

Conclusions
Capability - Quality Matrix



Current Issues &
Conclusions
Design to Cost

What can I develop for $X

rather than

How much will it cost to develop Y Program

Current Issues & Conclusions
COTS/GOTS Integration

TWO APPROACHES:

1) Estimate LOC needed for integration
NOTE: Do not run SEER SEM with LOC < 5,000

2) Find out Technical Information for each COTS package
Run SEER SEM with the information and only use the integration portion of the estimate

Conclusions Automated Code Generators

- Development Phase
  - Estimate only the code that is written by programmers
- Maintenance Phase
  - Estimate using the total delivered code

Current Issues & Conclusions
Commercial Cost Models

- Many of the technical parameters are unknown upfront
- 4GL and Object Oriented are not in the current database
- Statistically - Degrees of Freedom are not what you would like
- Extremely time consuming
  - For the engineer providing the information
  - For the estimator